

Université Assane Seck de Ziguinchor

UFR sciences et technologies

Département d'informatique



Mémoire de fin d'études

Pour l'obtention du diplôme de master

Mention : Informatique ; Spécialité : Génie Logiciel

Sujet :

**Automatisation de la gestion administrative du département
d'informatique de l'université Assane Seck de Ziguinchor
(AGADI-UASZ)**

Présenté par :

M^{elle} Ndickou Diop

Membres du jury

- Pr. Salomon SAMBOU (Directeur)
- D^r Ibrahima DIOP (**Rapporteur**)
- D^r Khadim DRAME (**Examineur**)
- D^r Youssou DIENG (Encadreur)

2018-2019

Résumé

Le département d'informatique, comme tous les départements de l'UASZ occupe une place très importante sur les activités pédagogiques et de recherche de l'université. La gestion du département est une tâche assez complexe et elle prend en charge la coordination de toutes ses activités. Le chef de département est chargé de la coordination et il a beaucoup de tâches à sa charge telles que la gestion des absences (Autorisation d'absence, Ordre de mission), du budget, des matériels alloués au département ainsi que les enseignants et leurs étudiants. Il est aidé dans ce travail par une assistante.

Pour mieux gérer ces tâches, le Chef du département d'informatique a jugé utile de mettre en place une application pour la gestion du dit département. Cette application nommée AGADI-UASZ permet, comme son nom l'indique, l'automatisation de la gestion administrative du département d'informatique de l'UASZ. Elle permet de :

- Dématérialiser la gestion des demandes d'autorisation d'absences, d'ordres de mission, les demandes de subvention etc. ;
- gérer automatiquement le budget du département;
- gérer l'historique des demandes et des activités du département ;
- gérer les dotations au niveau du département ;
- gérer le personnel ;
- gérer la programmation des rencontres (réunions) du département.

Pour mettre en place cette application, nous nous sommes proposés d'utiliser le processus 2TUP. Une spécification claire des besoins a permis de formaliser les données avec UML (Unified Modeling Language) et de les stocker dans MySQL (My Structured Query Language). Une implémentation avec PHP objet nous a permis de disposer d'une application fiable et optimale qui répond aux besoins du chef de département.

Abstract

The computer science department, like all departments of UASZ occupies a very important place on the educational and research activities of the university. The management of the department is a rather complex task and it takes care of the coordination of all its activities. The head of department is in charge of the coordination and he has many tasks at his charge such as the management of the absences (Authorization of absence, Order of mission), the budget, and the materials allocated to the department as well as the teachers and their students. He is helped in this work by an assistant.

To better manage these tasks, the Head of the IT department has found it useful to set up an application for the management of the said department. This application named AGADI-UASZ allows, as its name indicates, the automation of the administrative management of the computer department of the UASZ. She allows to:

- Dematerialize the management of absence authorization requests, mission orders, grant applications, etc. ;
- automatically manage the department's budget;
- manage the history of requests and activities of the department;
- manage departmental endowments;
- manage the staff;
- manage the scheduling of meetings (meetings) of the department.

To set up this application, we proposed to use the 2TUP process. A clear requirement specification has formalized the data with UML (Unified Modeling Language) and stored in MySQL (My Structured Query Language). An implementation with PHP object has allowed us to have a reliable and optimal application that meets the needs of the department head.

Dédicace

Par la grâce d'Allah, le tout puissant je dédie ce travail...

A mon père

C'est grâce à toi que nous sommes là aujourd'hui. Tu n'as ménagé aucun effort pour nous mettre dans des conditions exemplaires. Tu es et tu resteras le meilleur père.

Trouve dans ce travail l'expression de mon amour filial.

A ma mère

Les mots ne sont jamais assez forts pour exprimer la reconnaissance d'une fille à sa mère. Tu es pour nous une mère exemplaire qui a su nous entourer d'un grand amour sans nous faire oublier le respect de soi-même et l'aide à son prochain.

Je te dédie ce travail du fond du cœur en espérant que tu pourras être un jour fier de moi.

Tous mes vœux de santé, de bonheur et de longue vie.

A mes frères et sœurs

Puissions-nous rester unis solidaires dans la vie et fidèles à l'éducation que nos chers parents ont su nous inculquer.

Que Dieu continue à guider vos pas.

A ma famille d'accueil

Sans vous je ne saurais arriver là où je suis. Merci pour vos soutiens, vos conseils partagés et votre considération.

Aux familles DIOP, SARR, MBODJ, DIONE ET NIANE.

Les mots ne sont jamais forts pour exprimer ce que vous représentez pour moi. Soyez assurés de mon indéfectible attachement et de ma permanente disponibilité.

Au Dr Youssou DIENG

Nous vous avons connu tout au long de cette formation ; votre compétence, votre rigueur, vos qualités intellectuelles et humaines et votre simplicité forcent le respect et l'admiration de tous vos étudiants.

Nous vous prions de trouver ici le témoignage de notre vive reconnaissance.

Remerciements

Je remercie ALLAH le Tout-Puissant de m'avoir permis de mener à terme ces projets qui sont pour moi le point de départ d'une merveilleuse aventure, celle de la recherche, source de remise en cause permanente et de perfectionnement perpétuel.

Je tiens à remercier chaleureusement et à témoigner toute ma reconnaissance au Dr Youssou DIENG pour avoir bien voulu superviser ce modeste travail et donner de son temps et de son intelligence à la réussite de ce projet. Il représente pour moi un modèle de réussite et une source de motivation permanente, pour sa disponibilité, et son sens aigu de l'humanisme.

Nos vifs remerciements vont également aux membres du jury notamment le Professeur Salomon SAMBOU, le Docteur Ibrahima DIOP et le Docteur Khadim DRAMÉ pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Je tiens également à remercier le corps professoral de l'Université Assane Seck de Ziguinchor, non seulement pour vos précieux conseils et vos orientations, mais aussi pour votre disponibilité. Votre sensibilisation à la recherche et à l'innovation m'ont aidée à la réalisation de ce travail.

Mes remerciements s'adressent à mes parents, mon père Boubacar DIOP et ma mère Baffa SARR, qui m'ont inculqué un esprit de combativité et de persévérance et qui m'ont toujours poussée et motivée dans mes études. Sans eux, certainement je ne serai pas à ce niveau. Merci à vous Monsieur et madame Dione ainsi qu'aux petites Marième, Arame et Fatima qui sont devenues ma nouvelle famille depuis mon intégration à l'UASZ.

Merci également aux familles MBODJ, SARR et DIOP pour le soutien et l'encouragement qu'ils m'ont apporté tout au long de ce travail.

Par la même occasion, je remercie également tous les camarades étudiants pour tout le soutien qu'ils n'ont pas manqué de m'apporter. Il s'agit notamment des étudiants l'UFR ST, et des collègues de YPOS. Mention spéciale à Ismaila DIALLO, Camir MALACK, Ana BAKHOUM, Ndeye Madjiguène NDIAYE Babacar DIATTA du CRI, Ibrahima Mare et Maladho NDIAYE.

Un grand merci à Aboubakrine NDIAYE, Alassane SÈNE et M. DioneCounda YOCK pour leur soutien moral et financier.



Sommaire

Table des matières

<i>Résumé</i>	ii
<i>Abstract</i>	iii
<i>Dédicace</i>	iv
<i>Remerciements</i>	v
<i>Sommaire</i>	6
<i>Acronyms</i>	9
<i>Liste des tableaux</i>	11
<i>Liste des figures</i>	12
<i>Introduction Générale</i>	13
<i>Chapitre I : Présentation générale</i>	16
1. <i>Présentation de l' Université (UASZ)</i>	17
2. <i>Problématique</i>	20
3. <i>Solutions</i>	21
3.1. <i>Gestion des enseignants permanents</i>	21
3.2. <i>Gestion des enseignants non permanents</i>	22
3.3. <i>Gestion des enseignements et des étudiants</i>	22
3.4. <i>Gestion des rencontres du département, du patrimoine et de l'historique</i>	23
4. <i>Présentation du processus de développement</i>	23
<i>Conclusion</i>	25
<i>Chapitre II : Capture et analyse des besoins fonctionnels du système</i>	26
1. <i>Capture des besoins fonctionnels du système</i>	27
1.1. <i>Identification des acteurs du système</i>	27
1.2. <i>Identification des fonctionnalités du système d'authentification</i>	28
1.3. <i>Diagrammes des cas d'utilisation « use case »</i>	29
1.3.1. <i>Diagramme de cas d'utilisation de la gestion des utilisateurs</i>	30
1.3.2. <i>Diagramme de cas d'utilisation de la gestion du personnel</i>	30
1.3.3. <i>Diagramme de cas d'utilisation de la gestion du budget</i>	30
1.3.4. <i>Diagramme de cas d'utilisation de la gestion des absences</i>	31
1.3.5. <i>Diagramme de cas d'utilisation de la gestion des matériels</i>	31
1.4. <i>Description des cas d'utilisation</i>	31
2. <i>Analyse des besoins fonctionnels du système</i>	34

2.1. Les activités de l'authentification.....	34
2.2. Les activités d'ajout d'un utilisateur.....	34
2.3. Les activités de modification d'un utilisateur.....	35
2.4. Les activités de validation ou de rejet d'une demande.....	36
2.5. Diagrammes de séquences.....	36
2.5.1. Les séquences de l'authentification.....	37
2.5.2. Les séquences du cas « ajouter utilisateur ».....	37
2.5.3. Les séquences du cas « Modifier utilisateur ».....	37
2.5.4. Les séquences du cas « Supprimer utilisateur ».....	38
2.5.5. Les séquences de la « gestion d'une demande d'autorisation d'absence ».....	38
2.5.6. Les séquences du cas « modifier une demande ».....	39
Conclusion.....	39
Chapitre III : Capture des besoins techniques du système.....	40
1. Ressources disponibles au niveau du département.....	41
2. Architecture logicielle du système.....	41
2.1. Architecture 3-tiers.....	42
2.2. Architecture MVC.....	43
2.3. Serveur de base de données.....	44
2.4. Serveur web.....	45
Conclusion.....	46
Chapitre IV : Conception du système.....	47
1. Conception générique.....	48
1.1. Diagramme de déploiement.....	48
2. Conception préliminaire.....	49
2.1. Diagramme de composants du système.....	49
3. Conception détaillée du système.....	50
3.1. Dictionnaire de données.....	50
3.2. Diagrammes de classes.....	53
3.2.1. Diagramme de classes concernant la gestion des utilisateurs et leurs profils.....	54
3.2.2. Diagramme de classes concernant le reste du système.....	54
Conclusion.....	55
Chapitre V : Implémentation et Présentation du système.....	56
1. Implémentation du système de gestion du département d' informatique.....	57
1.1. Implémentation du SGBD(MySQL).....	57
1.1.1. Le modèle physique de données(MPD).....	57
1.1.2. La création de la Base de Données et des tables sous MySQL.....	57

2. Outils et technologies utilisés.....	58
2.1. Editeur de texte: Sublim Text.....	58
2.2. Outil de conception : Visual paradigm.....	59
2.3. Wamp Serveur.....	59
2.4 SGBDR MySQL.....	60
2.5. Outil de programmation PHP orienté Objet.....	60
3. Présentation de l' application.....	60
3.1. Présentation de l' interface de connexion.....	61
3.2. Présentation des pages d' accueil du système.....	61
3.2.1. Présentation de la page d' accueil du chef de département.....	61
3.2.2. Présentation de la page d' accueil de l' enseignant.....	62
3.2.3. Présentation de la page d' accueil de l' assistante.....	62
3.3. Scénario de gestion d' une demande.....	63
Conclusion.....	66
Conclusion générale et perspectives.....	67
Références.....	69
Annexes.....	71

Acronyms

2TUP:	Two Track Unified Process
AGADI-UASZ :	Automatisation de la gestion du département D'informatique de l'Université Assane Seck de Ziguinchor
BD:	Base de données
CSA	Chef des Services Administratifs
CSS:	Cascading Style Sheets
DA:	Diagramme D'activités
DCU:	Diagramme de Cas d'Utilisation
DS:	Diagramme de Séquences
Fk:	Foreign Key
HTML :	HyperText Markup Language
JS :	Java Script
MLD :	Modèle Logique de Données
MVC :	Modèle Vue Controller
MySQL:	My Structured Query Language
PATS :	Personnel Administratif Technique et de Service
PER :	Personnel Enseignant et Recherche
PHP :	Personal Home Page
PK :	Primary Key
RP :	Responsable Pédagogique
SGBD :	Système de Gestion des Bases de Données
SGBDR :	Système de Gestion des Bases de Données Relationnelles
SI:	System information
SQL:	Structured Query Language
UASZ :	Université Assane Seck de Ziguinchor
UFR :	Unité de Formation et de recherche
UML:	Unified Modelling Language
URL:	Uniform Resource Locator

XHTML : eXtensible HyperText Markup Langage
XP : eXtreme Programming

Liste des tableaux

Tableau 1: Liste des acteurs et leurs rôles.....	28
Tableau 2: Liste des fonctionnalités du système.....	29
Tableau 3: Description du cas Authentification	32
Tableau 4: Description du cas créer utilisateur.....	32
Tableau 5: Description du cas modifier utilisateur	33
Tableau 6: Description du cas Supprimer utilisateur.....	33
Tableau 7: Description du cas Ajouter demande d' autorisation.....	33
Tableau 8: Description du cas Valider/rejeter une demande d' autorisation.....	33
Tableau 9: Dictionnaire de données.....	53

Liste des figures

Figure 1: DRH de l'UGB après l'incendie du Rectorat en 2018.	21
Figure 2: Processus de développement en Y	24
Figure 3: Correspondance 2TUP et UML[4].....	25
Figure 4: Diagramme de cas d' utilisation de la gestion du budget.....	30
Figure 5: Diagramme de cas d'utilisation des absences.....	31
Figure 6: Diagramme d'activité du cas « s'authentifier »	35
Figure 7: Diagramme d'activité du cas « Ajouter utilisateur ».....	35
Figure 8: Diagramme d'activité du cas « valider/rejeter une demande».....	36
Figure 9: Diagramme de séquence du cas « authentication »	37
Figure 10: Diagramme de séquence du cas « ajouter utilisateur ».....	38
Figure 11: Diagramme de séquence de la gestion d'une demande Autorisation d'absence	39
Figure 12: Architecture trois tiers.....	42
Figure 13: Rôle des éléments du MVC.....	43
Figure 14: Architecture générale du système	44
Figure 15: Diagramme de déploiement du système	49
Figure 16: Diagramme de composant du système.....	50
Figure 17: Diagramme de classe concernant les utilisateurs et leur espace de travail	54
Figure 18: Diagramme de classes concernant le reste du système.....	55
Figure 19: Modèle physique de données du système gestion du département d' informatique	57
Figure 20: Création de la base de données du système gestion du département d' informatique	58
Figure 21: Création des tables de la base de données du système.....	58
Figure 22: Page de connexion.....	61
Figure 23: Page d'accueil du chef de département.....	62
Figure 24: Page d'accueil de l'enseignant	62
Figure 25: Page d'accueil de l'assistante du chef de département.....	63
Figure 26: Création d'une nouvelle demande d'autorisation d'absence	64
Figure 27: Page de gestion des demandes d'absence	64
Figure 29: Ajout d'une nouvelle demande de subvention	65
Figure 28: Page de la gestion des ordres de mission.....	66

Introduction Générale

La difficulté à accéder aux données physiques ne cesse d'être un problème pour les structures. En effet, avec les données physiques, les structures sont souvent exposées à des pertes de données en cas de sinistre qui peut être un incendie, une inondation... Une telle catastrophe dans les locaux d'une entreprise peut suffire à détruire en quelques minutes tous les documents papiers ou supports dans lesquels sont stockés des archives.

A l'UASZ, les départements sont souvent confrontés à des problèmes de perte et/ou d'accès aux documents au moment souhaité. En effet, un document mal rangé dans un tiroir peut être difficilement retrouvable et entraîner des dysfonctionnements graves dans la gestion du département. Le temps mis pour chercher et retrouver des documents peut aussi être préjudiciable pour toute entreprise.

Les chefs de départements mènent plusieurs activités en plus de la charge de travail administrative (la gestion des absences, la gestion du budget, la gestion des matériels). En effet, en plus de sa charge administrative de chef de département, il doit s'acquitter de sa charge d'enseignement qui est la même que celle de ses autres collègues. Il a aussi les mêmes obligations d'encadrement pédagogique que ses collègues de département.

Dans sa mission de chef de département, il est chargé de la coordination de toutes les activités du département. Il représente le département dans toutes les rencontres et réunions où le département est convié. Il vise tous les courriers administratifs qui partent des collègues du département. Il centralise aussi les courriers arrivant vers les collègues avant de les retransmettre. Il est chargé de répondre à toute sollicitation venant de la hiérarchie. Il prépare les réunions de département et vise et transmet les PV à la direction. Il prépare et présente le budget au conseil de département. Il est chargé de veiller à la bonne répartition du matériel et des équipements. Il est chargé de gérer les absences pour la bonne marche du département. En effet, en cas de déplacement, chaque enseignant-chercheur doit prendre une autorisation d'absence à moins d'avoir un ordre de mission. Il centralise les autorisations d'absence, et met son visa avant d'envoyer à la hiérarchie. En cas de déplacement d'un PER dans le cadre d'une mission, le Chef de département est chargé de rédiger une demande d'ordre de mission au Directeur d'UFR pour ce collègue.

Le PER comme le PATS du département peuvent aussi bénéficier de subventions ainsi que des dotations de matériels. Pour bénéficier d'une subvention, un enseignant-chercheur doit rédiger une demande motivée et adressée au Chef de département. Le Chef de département est chargé de vérifier la conformité des demandes avant de les transmettre au Directeur d'UFR

avec une lettre spécifiant le montant de la subvention accordée et la ligne qui doit être débitée sous réserve que celle-ci soit suffisamment dotée pour prendre en charge cette subvention. Il doit aussi informer l'intéressé de la suite donnée à sa demande.

L'exécution des tâches n'est pas sans difficulté car toutes les informations dont le chef de département dispose sont sur des supports papiers. On observe une lenteur dans le traitement des dossiers. Il est encore plus difficile de rechercher des données dans les archives. Ces données font souvent objet de perte. Cette façon de gérer l'information est inefficace. Aujourd'hui avec les technologies de l'information et de la communication (TIC), il existe des solutions innovantes qui gèrent de façon automatique et optimale toutes ces tâches.

Pour pallier aux difficultés, le département d'informatique a décidé de concevoir et d'implémenter un système d'information optimale et sûr pour l'automatisation de la gestion administrative du département.

Pour la mise en œuvre de cette application, le fonctionnement du département d'informatique est pris comme modèle de base. De l'étude de son fonctionnement, se sont découlées quelques idées qui ont permis de proposer les fonctionnalités du système de gestion du département que nous présentons ici.

Nous avons utilisé UML (Unified Modeling Language) pour formaliser les données qui seront stockées dans une base de données. Et, pour permettre l'interaction avec ces données, nous avons travaillé avec des technologies d'informatiques adaptées qui nous ont permis d'atteindre nos objectifs et mener à bien notre travail.

Nous avons réparti notre mémoire en cinq chapitres qui traitent successivement la **présentation générale** (Chapitre I) puis la **capture et l'analyse des besoins fonctionnels du système** (Chapitre II) ensuite celle des **besoins techniques du système et la Conception générique** (Chapitre III) et enfin on fera la **conception** (Chapitre IV) avant **d'implémenter et de présenter le système** (Chapitre V).

Chapitre I : Présentation générale

Dans un monde où la connectivité et la mobilité vont de pair, la rapidité d'accès à une information va de plus en plus devenir une exigence. Ainsi, les archives physiques vont de plus en plus engendrer des problèmes entre les clients et les entreprises. D'autres problèmes sont liés à la sécurisation des documents. A cela s'ajoute la difficulté à maîtriser les durées de conservation et les conditions de destruction des archives.

C'est la raison pour laquelle, dans le cadre de sa bonne gestion, toute structure a besoin de ressources informatiques. Cette ressource permettra d'avoir un accès automatique sur les données de la structure. En effet les données sont des informations propres à toute structure. Avec cet outil d'informatique, la recherche de données sera rapide. Ce qui produit un gain de temps et une efficacité dans le travail. C'est pour toutes ces raisons que le département d'informatique a jugé nécessaire de mettre en place une application de gestion du dit département. En effet, le département d'informatique comme la plupart des structures administratives de l'UASZ fonctionne toujours avec une gestion manuelle sous forme de documents physiques. Cette gestion engendre souvent des pertes de documents, une lenteur dans le traitement, une inefficacité dans le partage d'information, etc.

Pour mieux présenter le sujet, ce premier chapitre porte sur la description du problème et de la structure. Dans un premier temps, il aborde la présentation de l'université. Dans un deuxième temps, il décrit les problèmes de gestion que rencontre le département. Dans la troisième partie nous allons poser la problématique avant de faire la présentation du modèle de procédé que nous avons utilisé.

1. Présentation de l'Université (UASZ)

Créée par le Décret 2008-537 du 22 mai 2008, l'Université Assane SECK de Ziguinchor (UASZ) a ouvert ses portes en février 2007 avec un effectif de 257 étudiants répartis dans trois Unités de Formation et de Recherche (UFR) qui sont : l'UFR des Sciences et Technologies, l'UFR des Sciences Economiques et Sociales et l'UFR des Lettres, Arts et Sciences Humaines. Cette jeune université est implantée dans le quartier périphérique de Diabir, dans la commune de Ziguinchor.

Selon ce même décret, l'université Assane Seck constitue le « lieu approprié pour la recherche fondamentale et la recherche du développement, pour la production et la diffusion de connaissances indispensables au Sénégal ».

L'Université Assane SECK de Ziguinchor contribue fortement à valoriser les atouts économiques et culturels de la Casamance, qui, non seulement recèle d'importantes ressources culturelles et humaines, mais aussi constitue, eu égard à sa position géographique, un axe stratégique d'intégration sous régionale.

Pour sa part, l'institution a mis en place des filières qui répondent à la vocation agricole et touristique de la Casamance. Il s'agit notamment des filières Tourisme et Agroforesterie. Il a ouvert également des filières professionnelles telles que le Droit des affaires, l'informatique, etc. qui devront fournir des ressources humaines de qualité aux nombreux projets et programmes de la région. La volonté des autorités académiques de monter en puissance s'est matérialisée par la récente création de l'UFR des Sciences de la Santé (année académique 2011), en plus des trois UFR déjà existantes. Néanmoins, l'Université Assane Seck de Ziguinchor devra contribuer, à travers ses missions d'enseignement, de recherche et de services en faveur de la communauté, au développement de la sous-région. L'université se veut ainsi un pôle de formation et de recherche de qualité ouvert sur les réalités socio-économiques locales, nationales, sous régionales et internationales, et qui ambitionne d'être un des moteurs du développement durable du Sénégal et de la sous-région.

Elle a pour mission de [1] :

- former des cadres supérieurs qualifiés du Sénégal et des autres pays africains, principalement de la sous-région, dans des filières correspondant aux besoins de l'économie locale, nationale et régionale;
- développer une recherche scientifique de qualité en adéquation avec les problématiques locales et nationales ;
- privilégier la bonne gouvernance universitaire pour une utilisation optimale des ressources humaines, matérielles et financières de l'institution ;
- Utiliser les technologies de l'information et de la communication pour améliorer ses performances dans l'enseignement, la recherche et la gestion ;

Notre étude est centrée en premier lieu sur l'unité de formation et de recherche des Sciences et technologies plus particulièrement le département d'informatique. L'UFR a pour missions :

- le développement des activités d'enseignement et de recherche dans les domaines des mathématiques, de l'informatique, des sciences physiques, de la chimie et de l'agroforesterie ;
- la formation de cadres spécialisés ;
- la fourniture de services de formation continue et d'expertise ;
- le conseil dans les domaines des sciences et des technologies.

Le département d'informatique est l'un des départements de l'UFR à côté des départements de Physique, Chimie, Mathématiques, Agroforesterie et Géographie. Le département d'informatique est constitué d'un chef de département, d'une assistante, d'un responsable de master, des enseignants et des étudiants. Pour être étudiant en d'informatique, il faut, dès la première année, être inscrit en Mathématiques-Physique-Informatique (MPI). Dès la licence 2, l'étudiant de MPI a la possibilité de choisir entre Mathématiques-Informatique (MI) et Physique-Informatique (PI). Le département d'informatique partage donc une filière avec les départements de mathématiques et de Physique. Il faut attendre la licence 3 pour pouvoir se spécialiser en d'informatique. Dès lors, la spécialisation d'informatique ne commence qu'à partir de la licence 3. Par ailleurs, depuis quelques années, avec la montée en puissance de l'Université de Ziguinchor, le département d'informatique a mis en place un master en Informatique de spécialité Génie Logiciel. Il est constitué de deux options qui sont :

- Conduite des Projets informatiques ;
- Architecture Logiciels Distribués[2].

Le chef de département a pour rôle :

- d'arranger les heures de cours suivant la disponibilité des professeurs dans certains départements ;
- de présider les réunions et partager les informations pédagogiques ;
- de centraliser toutes les demandes et processus administratifs venant des enseignants et des étudiants et au besoin les retransmettre à la hiérarchie ;
- de gérer le budget et l'historique de leurs départements respectifs ;
- de gérer les matériels alloués aux enseignants.

Il est aidé par une assistante.

2. Problématique

Depuis la création de l'université, le Chef de département d'informatique rencontre énormément de difficultés pour mener à bien sa mission. Il ne dispose pas d'outils d'informatiques pour l'automatisation des suivis des activités pédagogiques ou administratives. Il est ainsi exposé aux pertes de données et aux problèmes d'omissions de document. Bref le chef de département ne dispose pas de moyens pour coordonner efficacement les activités. De même il lui sera difficile de gérer les activités journalières du département. Le chef de département n'a pas de moyens pour savoir automatiquement l'historique des missions, des voyages d'études, des autorisations d'absence, des réunions, des courriers administratifs qui ont été faits. Il est même difficile dès fois de répondre par oui ou non si un enseignant est présent ou en voyage. Cela ne facilite pas le travail administratif au niveau du département.

Le chef de département est chargé de faire énormément d'activités et sans aucun outil d'informatique.

Bien qu'il existe des coffres forts, des armoires étanches capables de résister à un sinistre, le risque de perdre une grande partie des archives est élevé. Par ailleurs, les documents les plus confidentiels du département peuvent être physiquement sur un bureau ou dans un tiroir au moment d'un incident. En 2018, un incendie au Rectorat de l'UGB a réduit en cendre le bureau du DRH comme vous pouvez le constater sur la [Figure 1](#). Dans ce cas de figure, à moins d'avoir un système informatisé avec un serveur de sauvegarde, la reconstruction de certains dossiers est impossible.

Si le département n'a pas défini une bonne politique d'archivage, retrouver un document peut devenir un véritable casse-tête : départ d'un chef, reprise d'un dossier, déménagement des services, salle d'archives entièrement remplie...

Il faut souligner également le manque d'espace destiné à la conservation des archives.

Au département d'informatique comme dans tous les départements de l'université Assane Seck de Ziguinchor, on constate que le volume des documents gérés ne cesse d'augmenter.

Pour retrouver un document, on est confronté à une perte de temps. La gestion des demandes de subvention et d'autorisation d'absence, de mission et de voyages d'étude se font manuellement. De même il n'y a pas la possibilité de gérer automatiquement les matériels du département.

On note plusieurs déplacements entre les enseignants, le chef de département, l'assistante pour déposer des demandes. Ce qui entraîne parfois des absences sans autorisation.

Le partage d'information de même que le suivi des activités ne sont pas automatisés dans le département. La gestion du département est manuelle et elle pose de véritables difficultés. Compte tenu de tous ces problèmes, il s'avère nécessaire de mettre en place une application de gestion du département d'informatique de l'université



Figure 1: DRH de l'UGB après l'incendie du Rectorat en 2018.

3. Solutions

Nous souhaitons mettre en place une application pour aider le chef de département d'informatique dans ses différentes tâches :

3.1. Gestion des enseignants permanents

Pour ce faire il sera mis en place une base de données pour stocker les informations relatives à chaque enseignant (matricule, nom, prénom, adresse, numéro de téléphone, grade,

etc.). Avec cette application, le chef de département peut gérer les autorisations d'absence, les demandes de subvention, etc. Il peut également gérer les voyages d'études qui sont des missions de recherches que les enseignants font de manière périodique (2 ans). L'application permettra de détecter automatiquement les enseignants qui en ont droit.

L'enseignant permanent ayant effectué des heures supplémentaires peut via l'application générer automatiquement un dossier de déclaration. L'application doit permettre à chaque enseignant d'initier tous les processus administratifs et faire lui-même le suivi.

3.2. Gestion des enseignants non permanents

Les enseignants non permanents sont des vacataires et des professionnels qui interviennent pour la professionnalisation des enseignements en d'informatique. Les vacataires en d'informatique sont souvent constitués de doctorants résidents avec une mission d'enseignement sur une durée couvrant tout le semestre. A cet effet, le chef de département élabore au début de chaque semestre une mission couvrant tous les enseignements qui leur sont affectés sur un semestre.

Les professionnels sont quant à eux engagés pour des missions de courte durée (entre une semaine à quinze jours). Ils sont ainsi logés et nourris par l'université et pendant toute la durée du séjour.

A la fin de chaque mission, un récapitulatif des heures effectivement enseignées est fait par le chef de département avant de demander son payement.

Au niveau de l'application, nous aurons une base de données qui stockera les informations de l'enseignant non permanent (matricule, nom, prénom, adresse, numéro de téléphone, grade, origine, matière enseignée, etc.).

Le chef de département peut utiliser l'application pour gérer les heures de vacation, c'est-à-dire pour chaque vacataire il aura la possibilité de voir les heures effectivement faites et générer un dossier de déclaration à la fin de la mission.

3.3. Gestion des enseignements et des étudiants

Au début de chaque semestre, le chef de département procède au dispatching des enseignements en d'informatique pour toute l'université. Ce travail se fait actuellement avec des feuilles de calcul Excel.

Avec l'application le chef de département peut faire le dispatching après le choix des enseignants et proposer des emplois du temps suivant la disponibilité des uns et des autres.

On aura en évidence une base de données qui stockera les informations sur les formations (L1, L2, L3, M1, M2, etc.) et sur les étudiants du département (matricule, nom, prénom, adresse, numéro de téléphone, niveau, etc.). Nous aurons aussi la possibilité de suivre les étudiants après leur sortie. Le chef de département pourra aussi à partir de l'application gérer les associations d'étudiants et les amicales.

3.4. Gestion des rencontres du département, du patrimoine et de l'historique

L'AGADI-UASZ permettra de faire des alertes de réunion (via Doodle), la saisie, l'impression et la possibilité d'envoi de rapports et ordre du jour. Le chef de département grâce à l'application peut gérer les salles de son département et le matériel qui est alloué aux personnels du département. Il a également la possibilité de faire le suivi administratif automatique des dossiers. En ce qui concerne l'historique, l'application permettra de gérer les historiques comme les missions, les voyages d'études, les autorisations d'absence qui ont été faits.

4. Présentation du processus de développement

Pour parvenir à une bonne résolution des problèmes rencontrés au niveau du département, nous suivons la méthode 2TUP qui signifie « 2 Track Unified Process ». Cette méthode 2TUP apporte une réponse aux contraintes de changement continu imposées aux systèmes d'information des organisations. En ce sens, il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes.

La méthode 2TUP, illustrée par la Figure 2 suivante est composée de trois branches[3] que sont :

- la branche fonctionnelle qui permet de réaliser la capture et l'analyse des besoins fonctionnels du système. Cette branche est celle qui est en rouge dans la Figure 2 suivante ;
- la branche technique qui permettra d'énumérer les besoins techniques et les architectures matérielle et logicielle du système. Elle correspond sur la Figure 2 à la branche colorée en vert ;

- la branche conception et développement logiciel a permis de faire les conceptions préliminaires et détaillées du système. A ce niveau aussi nous ferons le codage et le test. C'est celle colorée en bleu dans la Figure 2 ;

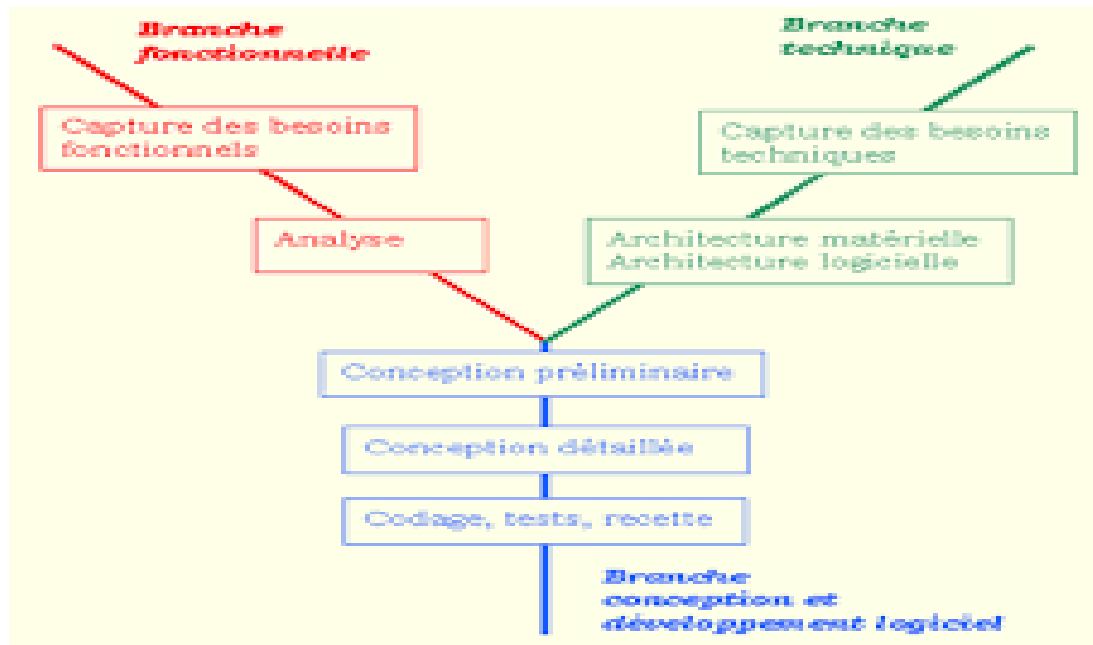


Figure 2: Processus de développement en Y

Ce processus suit une logique illustré par des diagrammes UML à chaque phase[4].

Comme le stipule la Figure 3 suivante :

- **capture des besoins fonctionnels** : au moment de la capture des besoins fonctionnels nous avons les diagrammes des cas d'utilisation, de séquences et de collaboration ;
- **analyse** : au moment de l'analyse nous avons les diagrammes de classes et de transition ;
- **capture des besoins techniques** : au moment de la capture des besoins technique nous avons le diagramme des cas d'utilisation ;
- **conception générique** : au moment de la conception générique nous avons le diagramme de déploiement ;
- **conception préliminaire** : au moment de la conception préliminaire nous avons les diagrammes de composant et de déploiement.
- **conception détaillée** : au moment de la conception détaillée nous avons les diagrammes de classes, de séquence, de collaboration, d'états, d'activités et de composition.



Figure 3: Correspondance 2TUP et UML[4]

Conclusion

En résumé ce chapitre a fait l'objet d'une description détaillée du problème. Il a permis de présenter notre structure d'accueil et même de l'institution à laquelle elle appartient. La problématique ainsi décrite dans sa globalité a conduit à la description des problèmes rencontrés par le chef de département. Il a ainsi permis d'exposer des attentes de l'application. Celles-ci sont jugées nécessaires pour une meilleure gestion du département d'informatique de l'université. Ce chapitre a aussi permis de présenter notre modèle de procédé. Ce qui permet d'entamer le chapitre 2 portant sur la capture et l'analyse des besoins fonctionnels du système.

*Chapitre II : Capture et analyse des
besoins fonctionnels du système*

Un système d'information est un ensemble organisé de ressources qui permet de collecter, regrouper, classer, traiter et diffuser de l'information sur un phénomène donné[5]. Concernant notre projet, nous considérons notre système comme étant le système de gestion du département d'informatique. Dans le cycle en Y, la capture et l'analyse des besoins fonctionnels constituent les deux (2) phases essentielles de la branche fonctionnelle. La première permet de comprendre les besoins de la structure. Cette compréhension induit évidemment à l'identification des acteurs, à la définition des fonctionnalités du système et à l'élaboration des diagrammes de cas d'utilisation. La seconde qui est la phase de l'analyse, elle permet de décomposer les besoins fonctionnels et d'élaborer les diagrammes qui permettent de la réussir.

Ainsi ce chapitre est consacré entièrement à la capture des besoins fonctionnels et à l'analyse du système de gestion du département d'informatique. Rappelons que notre structure d'accueil est le département d'informatique.

1. Capture des besoins fonctionnels du système

Les besoins fonctionnels sont l'ensemble des attributs portant sur l'existence d'un ensemble de fonctions et leurs propriétés données. Les fonctions sont celles qui satisfont aux besoins exprimés ou implicites.

La capture des besoins fonctionnels est la première étape de la branche gauche du cycle en Y. Elle permet de préciser l'étude du contexte fonctionnel du système en décrivant les différentes manières qu'auront les acteurs à l'utiliser. C'est une phase importante et obligatoire pour comprendre notre travail. Cette capture passe par l'identification des différents acteurs et la définition des fonctionnalités du système. L'élaboration des diagrammes de cas d'utilisation et de séquences est faite à ce niveau de même que la description des cas d'utilisation.

1.1. Identification des acteurs du système

Un acteur désigne une personne qui interagit avec le système. Il a un rôle à jouer et est interne ou externe. Un acteur peut être principal ou secondaire. Les acteurs représentent les utilisateurs du système. Pour connaître ces derniers, un questionnaire est fourni au chef du département d'informatique.

Suite aux réponses apportées à ce questionnaire, une liste des acteurs a été dégagée avec leurs rôles. Le Tableau 1 suivant représente la liste des acteurs et leurs rôles dans le système.

ACTEURS	Description	ROLES
Administrateur	C'est lui l'ingénieur qui se charge du système	Il assure la mise à jour du système : il a le privilège d'ajouter, de modifier ou de supprimer un utilisateur
Chef de département	Enseignant au département d'informatique	Il est le principal bénéficiaire des services offerts par le système. Il peut s'authentifier, modifier son mot de passe, faire une demande (autorisation d'absence, ordre de mission, subvention, etc.) ou la modifier. Il peut aussi ajouter un personnel, valider ou rejeter les demandes et/ou les envoyer au directeur UFR.
Assistante	Personnel administratif assistant du chef de département	Elle peut s'authentifier, modifier son mot de passe, faire une demande (autorisation d'absence, ordre de mission, subvention...),
PER	Personnel d'enseignement et de recherche	Le PER de l'université est aussi bénéficiaire des services offerts par le système. Ils peuvent s'authentifier, modifier leur mot de passe, faire une demande (autorisation d'absence, ordre de mission, subvention...),

Tableau 1: Liste des acteurs et leurs rôles

1.2. Identification des fonctionnalités du système d'authentification

L'étude préliminaire liste les besoins de la structure. En réponse aux besoins ou aux problèmes posés par le département d'informatique, un ensemble de fonctionnalités est identifié. Ils sont répertoriés dans le **Tableau 2** suivant :

Fonctionnalités du système	Acteurs
Modifier utilisateur	Administrateur
Supprimer utilisateur	Administrateur
Créer utilisateur	Administrateur
Modifier le mot de passe utilisateur	Chef du département, PER, Assistante, Administrateur
Faire demande d'autorisation d'absence	PER, Assistante, Chef département
Faire demande d'ordre de mission	PER, Assistante, Chef département
Faire demande de subvention	PER, Assistante, Chef département
Modifier demande d'autorisation d'absence	PER, Assistante, Chef département
Modifier demande d'ordre de mission	PER, Assistante, Chef département
Modifier demande de subvention	PER, Assistante, Chef département
Ajouter matériel	Chef département
Modifier matériel	Chef département
Allouer matériel	Chef département
Valider demandes	Chef département
Imprimer demandes	Chef département
Proposer date rencontres	PER, Assistante, Chef département
Consulter emploi du temps	PER
Afficher demandes	PER, Assistante, Chef département
Supprimer demandes	PER, Assistante, Chef département

Tableau 2: Liste des fonctionnalités du système

1.3. Diagrammes des cas d'utilisation « use case »

Un diagramme est une représentation graphique d'une séquence d'opérations ou de la structure d'un système.

Les diagrammes de cas d'utilisation sont des diagrammes UML utilisés pour donner une vision globale du comportement fonctionnel d'un système logiciel. Ils sont utiles pour des présentations auprès de la direction ou des acteurs d'un projet, mais pour le développement, les cas d'utilisation sont plus appropriés. Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec

un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service.

1.3.1. Diagramme de cas d'utilisation de la gestion des utilisateurs

L'administrateur gère les utilisateurs. Cette gestion passe par l'affichage, l'ajout, la modification ou la suppression d'utilisateurs. Pour gérer les utilisateurs, l'administrateur s'authentifie. Le diagramme de cas d'utilisation dudit système est représenté par l'[Annexe 1](#).

1.3.2. Diagramme de cas d'utilisation de la gestion du personnel

Le chef de département étant le principal utilisateur du système, accède à toutes les fonctionnalités. Cette gestion se fait par l'ajout, la modification, la suppression et l'affichage des utilisateurs. Pour ce faire, il doit s'authentifier.

Le diagramme de cas d'utilisation pour l'assistante et le chef de département est élaboré à l'[Annexe 2](#).

1.3.3. Diagramme de cas d'utilisation de la gestion du budget

Comme toute structure de bonne qualité, le département d'informatique a un budget qu'il gère. Ce budget est supervisé et organisé par le chef de département. Il est composé de la gestion des subventions, de son élaboration et des commandes faites au niveau du département.

La [Figure 4](#) suivante représente le diagramme de cas d'utilisation du budget.

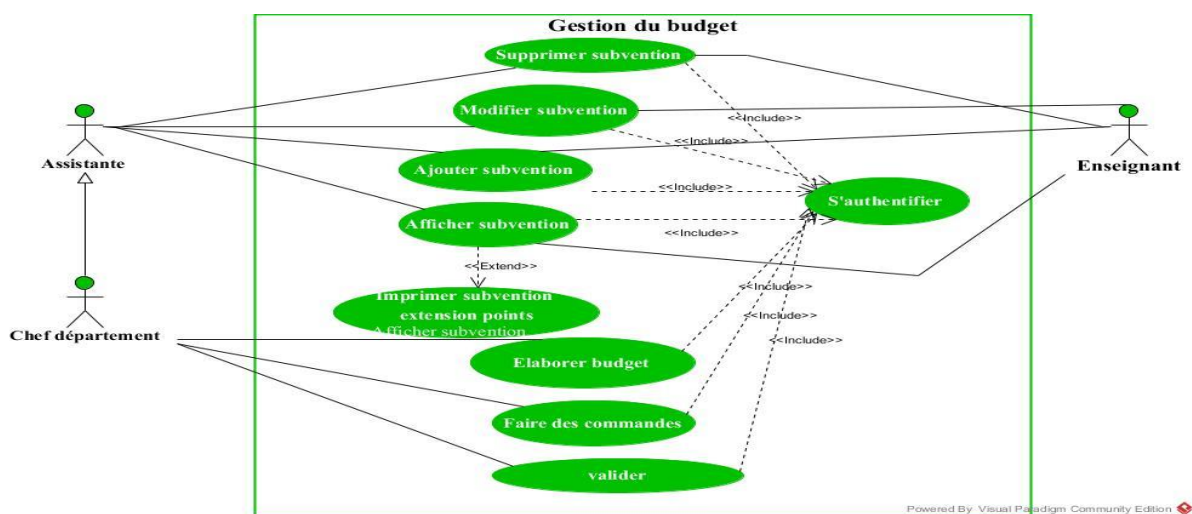


Figure 4: Diagramme de cas d'utilisation de la gestion du budget

1.3.4. Diagramme de cas d'utilisation de la gestion des absences

La gestion des absences fait partie du cœur de l'application. Elle est composée de la gestion des ordres de mission et celle des autorisations d'absence comme le montre la **Figure 5** suivante. A ce niveau tous les acteurs interviennent et peuvent tous faire des demandes d'autorisation d'absence ou d'ordre de mission. Par contre il ne revient qu'au chef de département de valider ou non les autorisations d'absences ou d'envoyer un ordre de mission aux hiérarchies.

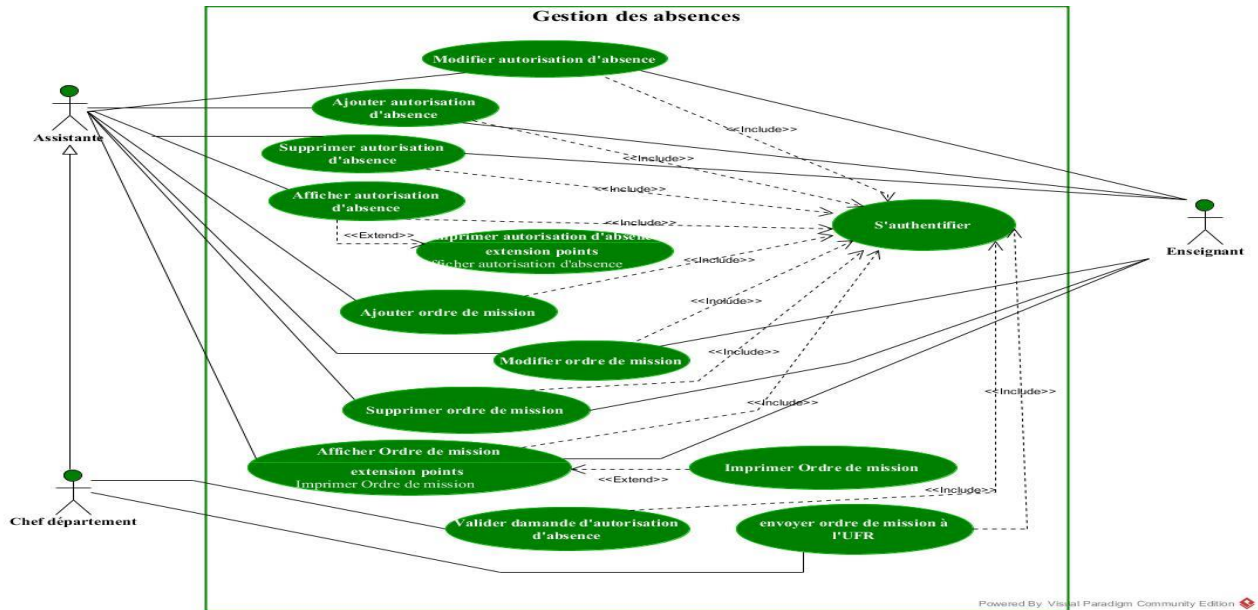


Figure 5: Diagramme de cas d'utilisation des absences

1.3.5. Diagramme de cas d'utilisation de la gestion des matériels

Au département d'informatique, des matériels, une fois arrivés au département, sont alloués à leurs propriétaires respectifs qui sont le personnel. L'**Annexe 3** suivante relate le diagramme de cas d'utilisation de la gestion des matériels. A ce niveau le chef de département ou l'assistante peuvent ajouter, modifier supprimer afficher ou allouer un matériel à un enseignant.

1.4. Description des cas d'utilisation

En génie logiciel et en ingénierie des systèmes, un **cas d'utilisation** définit une manière d'utiliser le système et permet d'en décrire les exigences fonctionnelles. Un cas d'utilisation est une unité cohérente représentant une fonctionnalité visible de l'extérieur. Il réalise un service de bout en bout, avec un déclenchement, un déroulement et une fin, pour l'acteur qui l'initie. Un

cas d'utilisation modélise donc un service rendu par le système, sans imposer le mode de réalisation de ce service[6].

Les tableaux suivants sont des descriptions de cas d'utilisation pour certaines fonctionnalités du système.

Titre	Authentification
Résumé	Ce cas permet de voir si l'utilisateur doit accéder au système ou pas. Et si oui, il le redirige vers les ressources auxquelles il a droit.
Acteurs	Administrateur, PER, Assistante, Chef département
Descriptions	A la tentative d'accès à l'application, une page web présentant l'interface d'authentification est affichée. l'acteur écrit son login et son mot de passe (on prendra ici l'email et le mot de passe de nos comptes dans www.univ-zig.sn
Préconditions	Etre PER au département d'informatique, assistante
Post conditions	On accède au système

Tableau 3: Description du cas Authentification

Titre	Créer utilisateur
Résumé	Créer un compte à un utilisateur
Acteur	Administrateur
Descriptions	S'il y a une nouvelle recrue, on lui crée un login et un mot de passe.
Préconditions	Authentification de l'administrateur
Post conditions	Le nouvel utilisateur est créé.

Tableau 4: Description du cas créer utilisateur

Titre	Modifier utilisateur
Résumé	L'administrateur modifie le login de l'utilisateur
Acteur	Administrateur
Descriptions	S'il y a erreur, l'administrateur modifie le login, les données ou le mot de passe en cas d'oubli de celui-ci.
Préconditions	Authentification

Post conditions	L'utilisateur est modifié dans le système
-----------------	---

Tableau 5: Description du cas modifier utilisateur

Titre	Supprimer utilisateur
Résumé	Enlever un utilisateur de l'annuaire
Acteur	Administrateur
Descriptions	En cas de limogeage, du décès, du transfert ou de l'abandon, l'administrateur supprime la personne
Préconditions	Authentification
Post conditions	Il n'est plus dans l'annuaire

Tableau 6: Description du cas Supprimer utilisateur

Titre	Ajouter demande d'autorisation (absence, mission, subvention.)
Résumé	Faire des demandes d'autorisation
Acteur	Chef du département, enseignants, assistante
Descriptions	Si l'authentification est réussie, l'acteur fait des demandes d'autorisation
Préconditions	Authentification
Post conditions	La demande est envoyée au niveau de l'espace de travail du chef département

Tableau 7: Description du cas Ajouter demande d'autorisation

Titre	Valider/rejeter une demande d'autorisation
Résumé	Entrer dans une demande la valider ou la rejeter
Acteur	Chef département
Descriptions	Si la connexion est réussie, le chef de département visualise la demande et l'accorde à l'enseignant ou la rejette
Préconditions	Authentification
Post conditions	La demande est validée ou rejeter

Tableau 8: Description du cas Valider/rejeter une demande d'autorisation

La capture des besoins fonctionnels est une phase capitale dans le cycle de vie du logiciel (2TUP). Elle permet d'identifier les fonctionnalités du système ainsi que les différents

acteurs qui l'utilisent. Cette partie du chapitre montre l'identification des utilisateurs et les fonctionnalités du système. Celles-ci représentent les cas qui sont décrits dans le diagramme de cas d'utilisation.

2. Analyse des besoins fonctionnels du système

Dans le processus de développement d'un logiciel, l'analyse constitue une étape très importante. Elle consiste à définir les fonctions que le produit ou service devra remplir. Chacune des fonctions doit être qualifiée par un niveau de performance. Les fonctions seront hiérarchisées entre elles afin de préparer d'éventuels arbitrages si on ne peut pas techniquement ou financièrement toutes les réaliser. Cette analyse fonctionnelle est un travail conjoint entre les futurs utilisateurs ou leurs représentants et ceux qui vont concevoir le produit ou service. En effet la définition de certaines fonctions peut demander des connaissances techniques.

Dans cette partie, nous allons faire l'analyse du système en décrivant les activités des fonctionnalités les plus importantes de ce dernier et leurs séquences.

Ces fonctionnalités permettent de résoudre les problèmes liés à la gestion du département d'informatique.

2.1. Les activités de l'authentification

Le cas d'utilisation « s'authentifier » est composé de plusieurs activités. La première consiste à saisir un nom d'utilisateur et un mot de passe. Une fois que cette activité est exécutée, le système procède à une vérification. Si les données saisies sont incorrectes, un message d'erreur s'affiche et l'activité doit être reprise. Elle ne sera effective que si les données sont correctes.

Par ailleurs, si les champs sont bien renseignés, elle débouche sur l'accès à la page d'accueil. L'ensemble de ces enchaînements est décrit dans le diagramme d'activité de la Figure 6.

2.2. Les activités d'ajout d'un utilisateur

L'administrateur choisit d'ajouter un utilisateur, le formulaire d'ajout est affiché, il le remplit et le valide. Si les données sont correctes, l'utilisateur est enregistré sinon il reçoit un message d'erreur lui disant qu'il n'a pas bien rempli le formulaire. La Figure 7 suivante décrit ce diagramme d'activité du cas « ajout utilisateur ».

Il en est de même pour l'ajout d'une demande d'autorisation d'absence, de subvention ou de mission mais l'acteur concerné sera le chef de département.

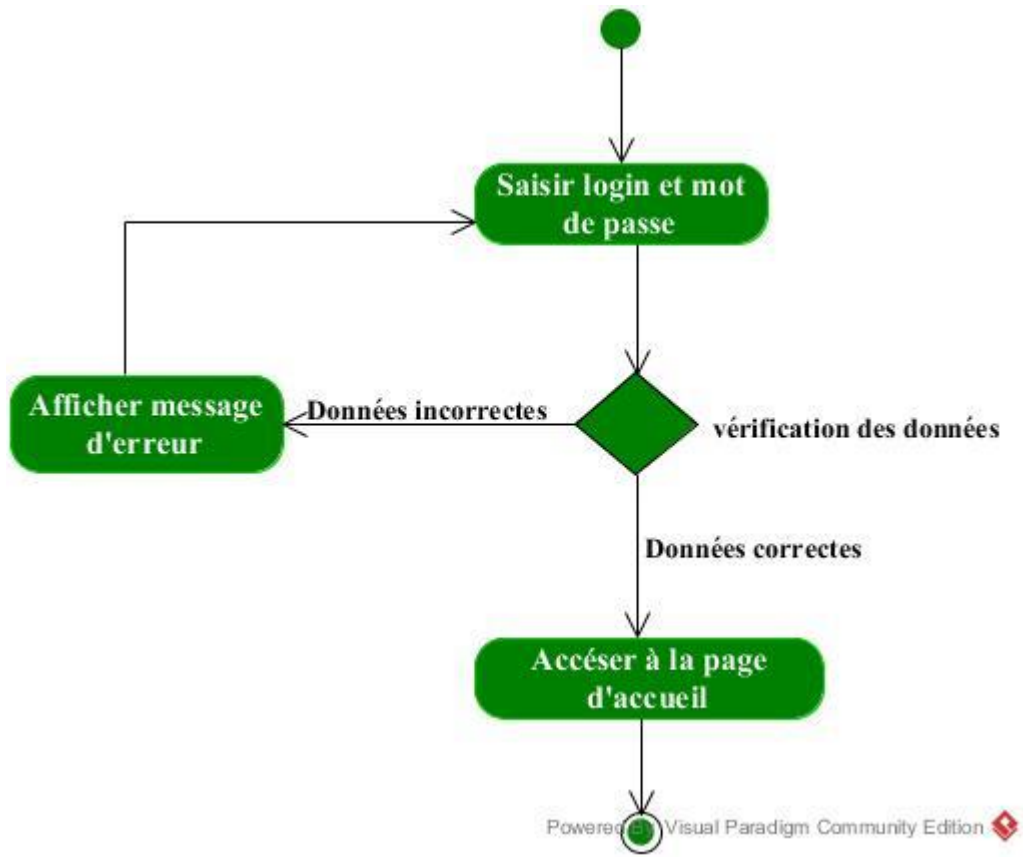


Figure 6: Diagramme d'activité du cas « s'authentifier »

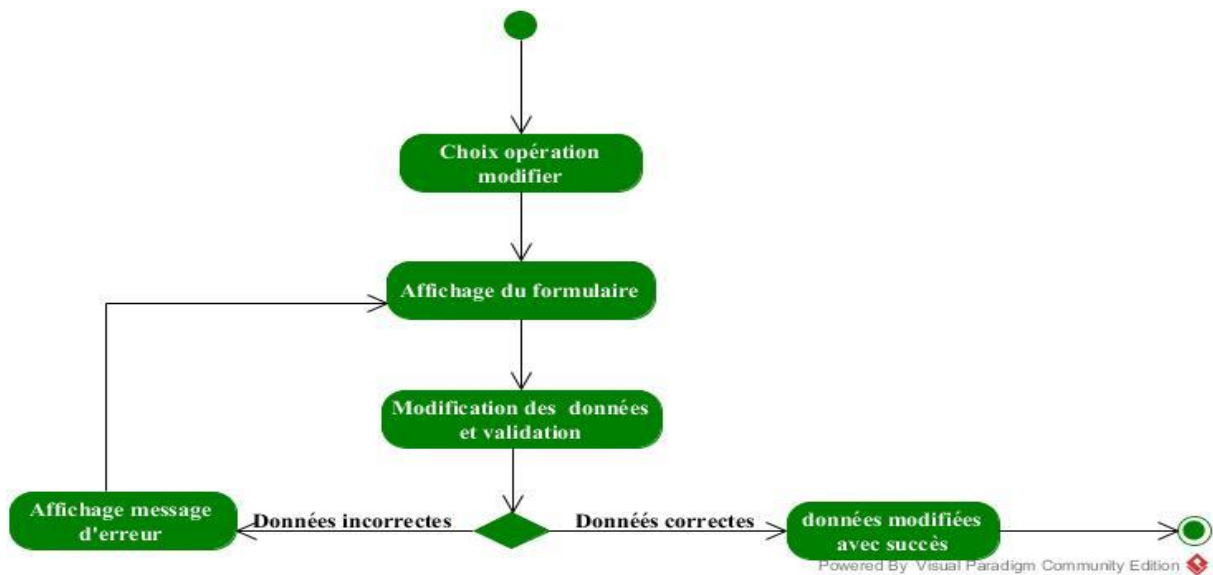


Figure 7: Diagramme d'activité du cas « Ajouter utilisateur »

2.3. Les activités de modification d'un utilisateur

L'administrateur choisi de modifier un utilisateur, le formulaire rempli est affiché, il le modifie et le valide. Si les données sont correctes, l'utilisateur est modifié sinon il reçoit un

message d'erreur lui disant qu'il n'a pas bien rempli le formulaire. L'[Annexe 5](#) décrit ce diagramme d'activité du cas « modifier utilisateur »

Il en est de même pour la modification d'une demande d'autorisation d'absence, de subvention ou de mission mais l'acteur concerné sera le chef de département.

2.4. Les activités de validation ou de rejet d'une demande

Le chef de département, étudie la demande de l'enseignant. Deux cas sont possibles, soit il le valide et l'envoie à son demandeur ou il l'imprime, soit il le rejette et dans ce cas le système envoie un message à l'enseignant pour l'informer du rejet de sa demande. La [Figure 8](#) suivante décrit ce diagramme d'activité du cas « valider/rejet de la demande »

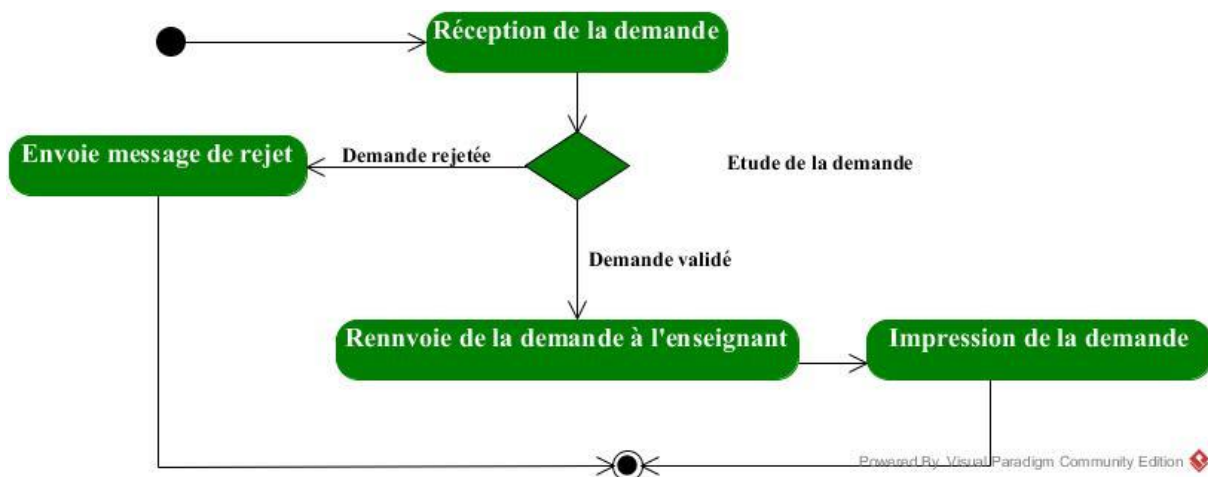


Figure 8: Diagramme d'activité du cas « valider/rejeter une demande »

Il en est de même pour l'ajout d'une demande d'autorisation d'absence, de subvention ou de mission mais l'acteur concerné sera le chef de département.

2.5. Diagrammes de séquences

Le diagramme de séquences fait partie des diagrammes comportementaux (dynamique) et plus précisément des diagrammes d'interactions. Il permet de représenter des échanges entre les différents objets et acteurs du système en fonction du temps. Nous ferons donc appel à un ensemble de diagrammes de séquences chacun correspondant à une sous fonction du système, généralement d'ailleurs pour illustrer un cas d'utilisation. Pour notre cas nous allons donner les diagrammes de séquences des cas d'utilisations tels que l'authentification, l'ajout, la suppression, la modification d'une demande ou d'un utilisateur en considérant les scénarios nominaux.

2.5.1. Les séquences de l'authentification

Nous remarquons qu'à chaque diagramme de cas d'utilisation il nous faut s'authentifier. La **Figure 9** **Erreur ! Source du renvoi introuvable.** suivante donne les séquences de ce cas d'utilisation.

Pour s'authentifier, l'utilisateur saisit le login et le mot de passe. Le système de base de données procède à une vérification. Une fois que cela est fait, l'utilisateur accède à la page d'accueil.

2.5.2. Les séquences du cas « ajouter utilisateur »

L'administrateur accède à la page d'accueil et décide d'ajouter un utilisateur. Il va dans l'onglet personnel et clique sur le bouton ajouter. Le système envoie le Formulaire et l'administrateur saisit les informations nécessaires pour l'ajout. Le système de base de données vérifie et procède à l'enregistrement. La **Figure 10** **Erreur ! Source du renvoi introuvable.** suivante nommée Diagramme de séquence du cas « ajouter utilisateur » traduit cette procédure.

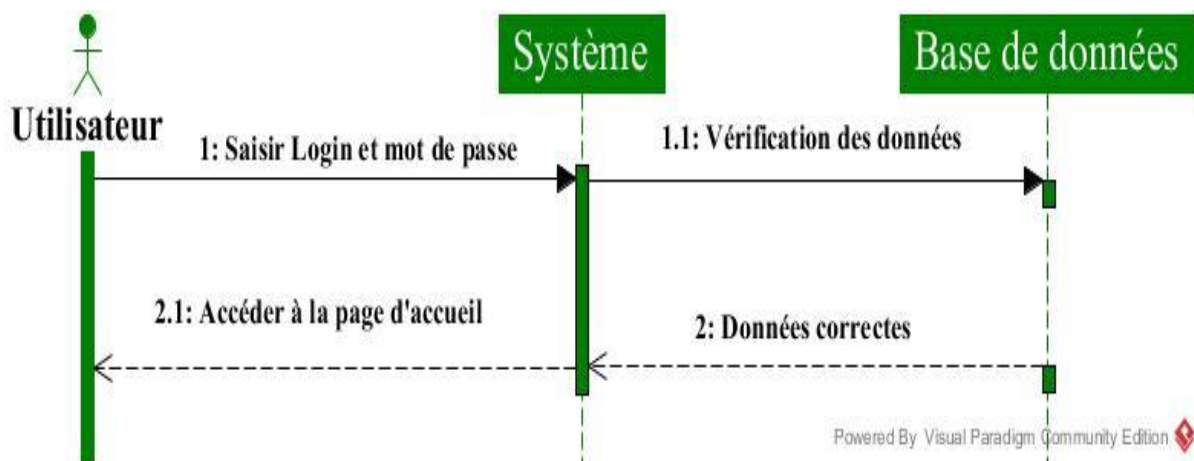


Figure 9: Diagramme de séquence du cas «authentification»

2.5.3. Les séquences du cas « Modifier utilisateur »

L'administrateur accède à la page d'accueil et décide de modifier les données concernant un utilisateur. Il choisit sur la liste des utilisateurs celui qu'il veut modifier et va sur le bouton modifier. Le système envoie le Formulaire rempli et l'administrateur modifie les informations. Le système de base de données vérifie et procède à l'enregistrement. L' **Annexe**

4 nommée Diagramme de séquence du cas « ajouter utilisateur » traduit cette procédure.

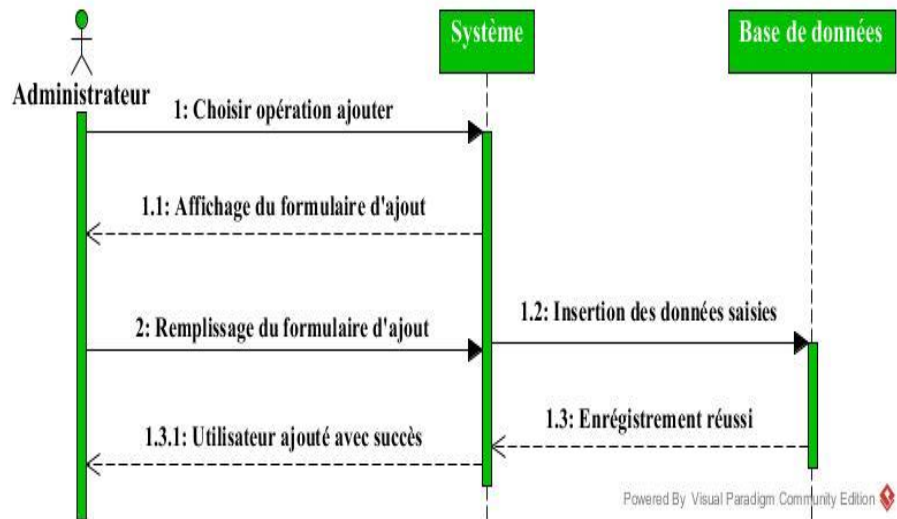


Figure 10: Diagramme de séquence du cas « ajouter utilisateur »

2.5.4. Les séquences du cas « Supprimer utilisateur »

L'administrateur accède à la page d'accueil et décide de supprimer un utilisateur. Il choisit sur la liste des utilisateurs celui qu'il veut supprimer et va sur le bouton supprimer. Le système envoie un message de confirmation et l'utilisateur est supprimé de la base de données.

2.5.5. Les séquences de la « gestion d'une demande d'autorisation d'absence »

Cette phase relate le cursus de la gestion d'une demande d'autorisation d'absence. Elle regroupe deux cas d'utilisation répartis en section dans le premier cas on a l'ajout d'une demande et dans l'autre il y a le cas de la validation de la demande.

L'enseignant accède à la page d'accueil et décide d'ajouter une demande d'autorisation d'absence. Un formulaire d'ajout est affiché, il remplit le formulaire et le système procède à une vérification des données puis il l'enregistre dans la base de données.

Le chef de département une fois connecté reçoit la demande et procède à sa validation ou à son rejet. La **Figure 11** nommée Diagramme de séquence de la gestion d'une demande d'Autorisation d'absence traduit cette procédure.

Cette séquence est de même pour l'ajout des demandes d'ordre de mission, de subventions.

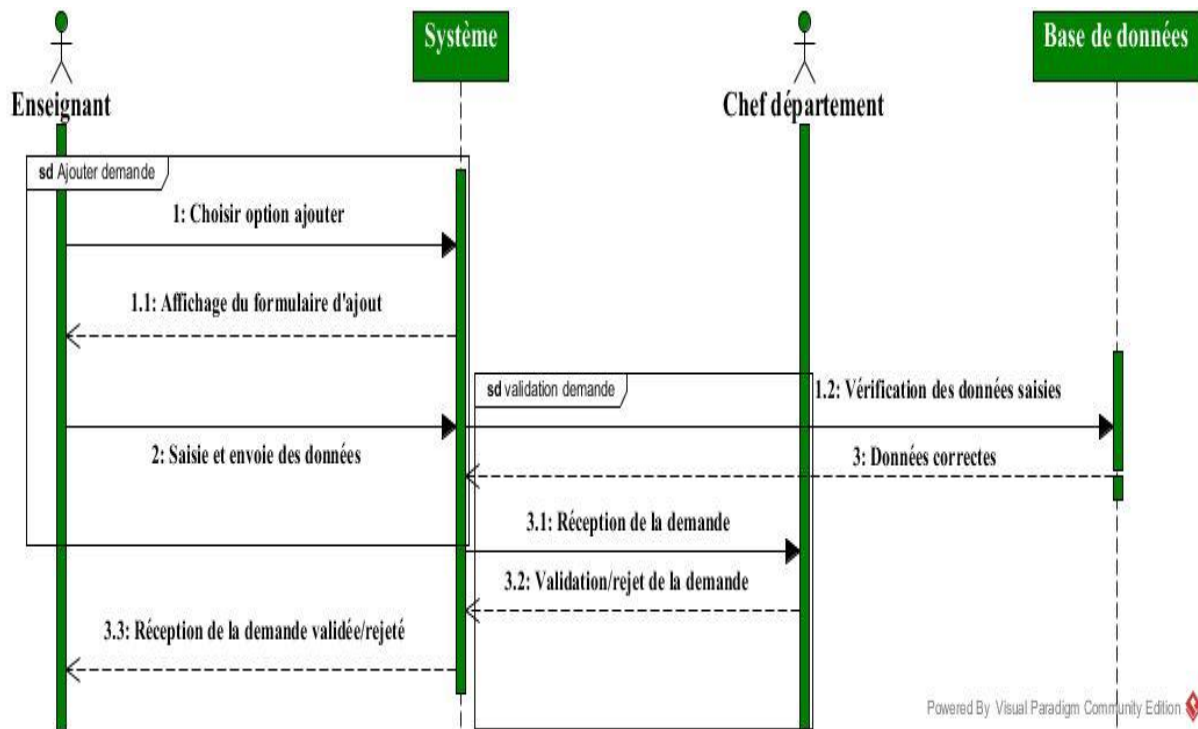


Figure 11: Diagramme de séquence de la gestion d'une demande Autorisation d'absence

2.5.6. Les séquences du cas « modifier une demande »

Le chef de département ou l'enseignant accède au système et décide d'opérer des mutations concernant une demande. Le système délivre le formulaire avec les champs rempli. Il opère et valide les modifications. Le système procède à une vérification des changements effectués et les intègre.

L'analyse des besoins fonctionnels du système a permis de mieux comprendre les interactions qui existent entre le système et les utilisateurs et les relations qui existent entre ces utilisateurs.

Conclusion

Ce chapitre a permis de faire la capture des besoins fonctionnels et l'analyse du système. Il a aussi permis d'énumérer les acteurs du système. Le chapitre suivant entame une autre étape importante de la méthode 2TUP qui est la capture et l'analyse des besoins techniques.

*Chapitre III : Capture des besoins
techniques du système*

La capture des besoins techniques détermine l'une des étapes de la branche technique (branche colorée en vert à la [Figure 2](#) de la méthode **2TUP**.

La capture des besoins techniques couvre, par complémentarité avec celle des besoins fonctionnels, toutes les contraintes qui ne traitent ni de la description du métier des utilisateurs, ni de la description applicative.

Le modèle des spécifications logicielles concerne donc les contraintes techniques. Cette spécification technique est une activité de la branche droite du Y; elle est primordiale pour la conception d'architecture

En effet ce chapitre doit montrer l'ensemble des choix techniques utilisés pour le développement du logiciel d'AGADI-UASZ. Ainsi, il est question de définir l'architecture logicielle et ces éléments.

Il s'agit d'un travail souvent moins décrit dans les méthodes d'analyse et il ne dispose pas de formalisation standardisée[7]. Son importance et son existence même dépendent de la complexité de l'application et de son environnement.

1. Ressources disponibles au niveau du département

Elles englobent l'ensemble des moyens matériels, humains et logistiques du département pour le système.

Les ressources humaines (les personnes) : pour le système de gestion du département, nous avons l'intervention du chef de département, l'assistante et le personnel enseignant du département (pers).

Le chef de département a la prérogative de valider ou de rejeter les demandes et de faire parvenir celles-ci au supérieur. L'assistante aide le chef de département dans son travail quant aux pers, ils font des demandes.

Les bureaux : le chef de département, le per et l'assistante ont leur propre bureau. Ils accèdent tous au réseau de l'université.

Les machines : chaque personne a sa machine. La majorité est donnée par le département. Les machines sont de 4Go de RAM, Pentium, 2.40GHZ, Windows 8, 64bits

Les réseaux : le département utilise le Wifi et le réseau filaire de l'Université. Aucune application d'informatique n'est utilisée.

2. Architecture logicielle du système

L'architecture logicielle consiste à décrire l'organisation générale d'un système et sa décomposition en sous-systèmes ou composants. Elle consiste aussi à déterminer les interfaces

entre les sous-systèmes, de décrire les interactions et le flot de contrôle entre les sous-systèmes[8]. Autrement dit, une architecture logicielle est une représentation abstraite d'un système exprimée essentiellement à l'aide de composants logiciels en interaction via des connecteurs. Il existe plusieurs types d'architectures tels que:

2.1. Architecture 3-tiers

L'architecture trois tiers, aussi appelée architecture à trois niveaux ou architecture à trois couches, est l'application du modèle plus général qui est le multi-tiers. L'architecture logique du système est en trois niveaux :

- la couche de présentations des données ou la couche client : correspondant à l'affichage, la restitution sur le poste de travail, le dialogue avec l'utilisateur ;
- la couche de traitement métier (serveur d'application) : correspondant à la mise en œuvre de l'ensemble des règles de gestion et de la logique applicative ;
- l'accès aux données persistantes (serveur de base de données): correspondant aux données qui sont destinées à être conservées sur la durée, voire de manière définitive [9].

Voici la [Erreur ! Source du renvoi introuvable.](#) qui le montre:

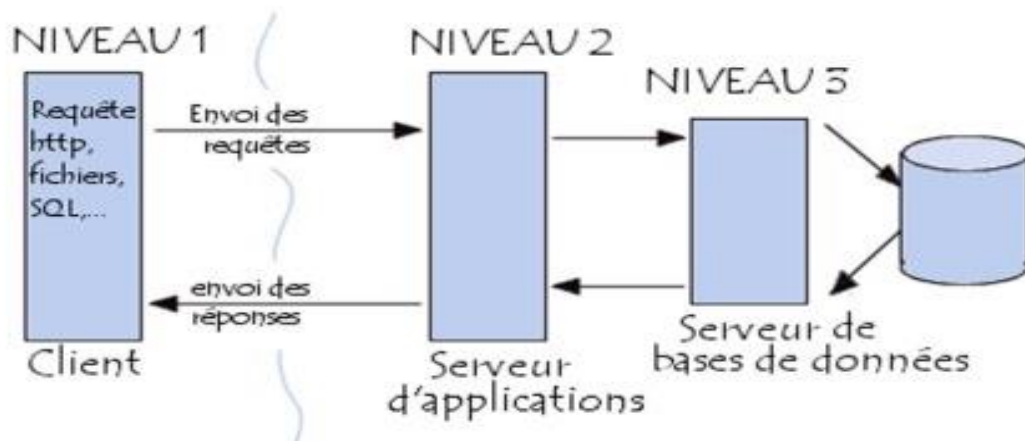


Figure 12: Architecture trois tiers

Pour mieux comprendre notre architecture, nous allons donner les différents éléments qui constituent le système :

1. un système de gestion des demandes (autorisation d'absence, mission, subvention...)
2. le client web qui va interagir avec le système ;
3. le serveur d'application ;
4. la base de données où nous allons stocker toutes les données ;
5. l'imprimante qui va permettre d'imprimer les demandes ;

2.2. Architecture MVC

Le pattern MVC (Model-View-Controller) permet de bien organiser son code source. Il va nous aider à savoir quels fichiers créer, mais surtout à définir leur rôle. Le but de MVC est justement de séparer la logique du code en trois parties que l'on retrouve dans des fichiers distincts, comme l'explique la description qui suit :

- **Modèle** : cette partie gère les données de votre site. Son rôle est d'aller récupérer les informations « brutes » dans la base de données, de les organiser et de les assembler pour qu'elles puissent ensuite être traitées par le contrôleur. On y trouve donc les requêtes SQL.
- **Vue** : cette partie se concentre sur l'affichage. Elle ne fait presque aucun calcul et se contente de récupérer des variables pour savoir ce qu'elle doit afficher. On y trouve essentiellement du code HTML mais aussi quelques boucles et conditions PHP très simples, pour afficher par exemple la liste des demandes.
- **Contrôleur** : cette partie gère la logique du code qui prend des décisions. C'est en quelque sorte l'intermédiaire entre le modèle et la vue : le contrôleur va demander au modèle les données, les analyser, prendre des décisions et renvoyer le texte à afficher à la vue. Le contrôleur contient exclusivement du PHP. C'est notamment lui qui détermine si le visiteur a le droit de voir la page ou non (gestion des droits d'accès)[8].

La Figure 13 suivante schématise le rôle de chacun de ces éléments.

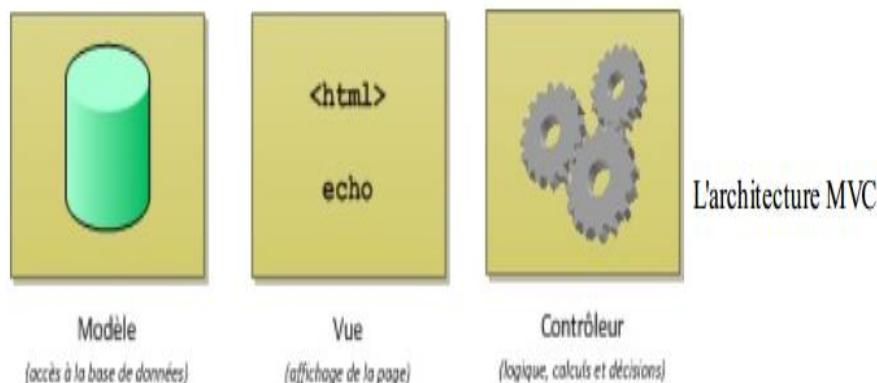


Figure 13: Rôle des éléments du MVC

La **Figure 14** suivante montre l'architecture générale de notre système en prenant en compte le client, le serveur d'application et la base de données. Elle montre également que le modèle MVC constitue notre serveur d'application.

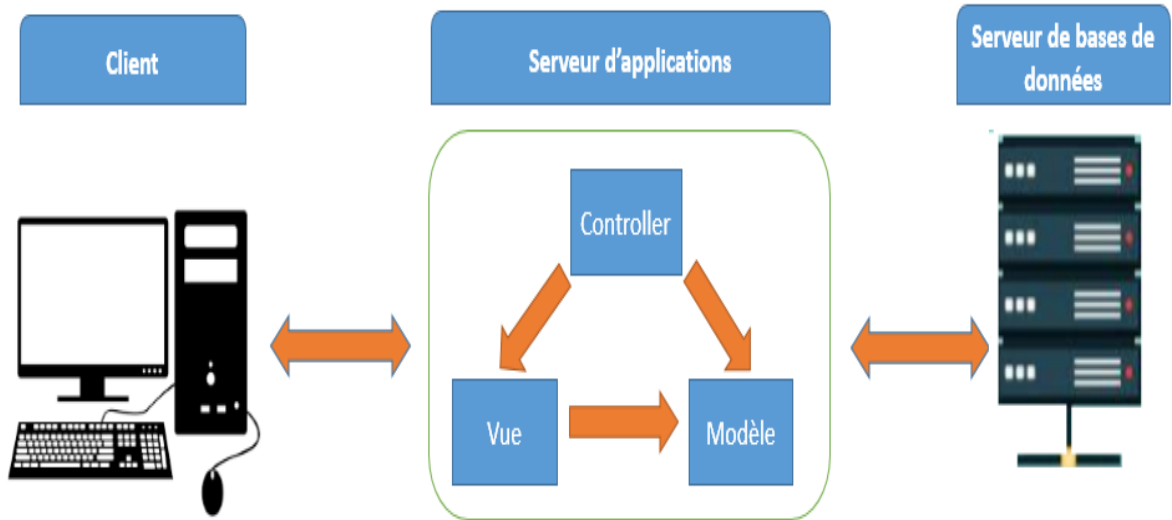


Figure 14: Architecture générale du système

2.3. Serveur de base de données

Une base de données (Database en anglais) est un ensemble organisé d'informations mémorisées sur un support d'informatique. Elle permet aux utilisateurs d'accéder aux données et de faire des mises à jour. Celles-ci dépendent des privilèges qui leur sont octroyés. Une base de données est locale ou centralisée. Ainsi pour pouvoir gérer une base de données, il est nécessaire d'avoir un SGBD (Système de gestion des bases de données). Un SGBD est un logiciel qui permet de créer et de gérer une base de données. Il facilite le traitement et la manipulation des données grâce à des requêtes. Ces dernières sont faites via des langages reconnus par les SGBD. Le plus populaire de ces langages est SQL. Databasic, dataflex, dBase ou xBaseScript sont aussi des langages de manipulation de SGBD. Les principaux SGBD sont : Microsoft Access, Microsoft SQL server, Oracle, MySQL, SQL server, PostgreSQL, Sybase, Teradata, Berkeley BD, Interbase, etc. Il existe plusieurs types de SGBD :

- Les SGBD relationnels (SGBDR) : selon les SGBDR, les données sont enregistrées dans des tableaux à deux(2) dimensions contenant des lignes et des colonnes. Ils sont appelés des tables. Les données contenues dans la base de données sont susceptibles de faire l'objet d'une recherche. Cette recherche est facilitée par le nom de la table dans

laquelle les données recherchées sont répertoriées. Les SGBDR sont les plus utilisés. Certains d'entre eux sont gratuits (open source) et d'autres ne le sont pas.

- Les SGBD hiérarchiques : les données sont représentées hiérarchiquement et suivent une arborescence descendante. Ils utilisent des pointeurs entre les différents enregistrements. Ce sont les premiers SGBD.
- Les SGBD en réseau : ils ressemblent aux SGBD hiérarchiques. Cependant l'arborescence ne peut pas être descendante.
- Les SGBD prédictifs : les données sont indiquées sous forme de tableaux mais leur manipulation se fait par calcul de prédicats.
- Les SGBD objets : c'est la nouvelle génération de SGBD. Les données sont représentées sous forme d'objets.

Les types de SGBD sont différents. Leur choix dépend de l'application et des bases de données à utiliser. Aujourd'hui, les SGBD relationnels sont les plus utilisés. MySQL est un SGBDR développé dans un souci de performances élevées en lecture. Ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multiutilisateur et multithread. C'est un logiciel développé sous double licence selon qu'il est distribué avec un produit libre ou avec un produit propriétaire. MySQL fonctionne sur de nombreux systèmes d'exploitation. Il supporte deux(2) langages d'informatiques, le langage SQL et le SQL/PSM (Persistent Stored Modules). Il fait partie des SGBDR les plus utilisés et est facile d'apprentissage [10].

2.4. Serveur web

Plusieurs serveurs web sont aujourd'hui utilisés. Chaque serveur présente des caractéristiques qui lui sont propres. Le choix d'un tel ou tel autre serveur réside dans le type de projet que l'on souhaite réaliser, aux moyens dont on dispose et des avantages qu'il fournit. Après avoir fait des recherches sur plusieurs serveurs web, le choix est porté sur apache de wamp.

WampServer est une plate-forme de développement Web sous Windows. Il nous permet de développer des applications Web dynamiques à l'aide du serveur Apache2, du langage de

scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement nos bases de données.

Contrairement aux autres solutions, WampServer nous permet de reproduire fidèlement notre serveur de production[11].

Conclusion

La capture des besoins techniques constitue l'une des étapes les plus importantes du cycle en Y et permet d'éclairer sur les aspects techniques du système. En effet, elle constitue le moment adapté au listing des choix et spécifications techniques. Toutes les ressources matérielles et humaines mises à la disposition du système de gestion du département sont connues. Aussi, ce chapitre a permis de voir les choix liés au SGBD (MySQL), à l'architecture logicielle, au langage (PHP) et au serveur web (apache) utilisés.

Au sortir de ce chapitre, les choix et spécifications techniques sont bien ficelés. La fin de ce chapitre nous permet de commencer une autre phase déterminante de 2TUP qui est celle de la conception.

Chapitre IV : Conception du système

La conception du système constitue la phase essentielle de la branche du milieu en plus de la conception générique qui est la deuxième partie de la branche de droite du processus **2TUP**.

La conception met en œuvre un ensemble d'activités qui à partir d'une demande d'informatisation d'un processus (demande qui peut aller de la simple question orale jusqu'au cahier des charges complet) permettent la celle-ci, l'écriture et la mise au point d'une application (et donc de programmes d'informatiques) jusqu'à sa livraison au demandeur[13] .

Tandis que la conception générique définit les composants nécessaires à la construction de l'architecture technique.

Dans la conception générique, les diagrammes de package et de déploiement doivent être élaborés.

Dans la troisième branche, on a la conception préliminaire qui nous permettra d'aborder le diagramme de composants du système et celle détaillée où le dictionnaire de données et le diagramme de classe sont érigés.

1. Conception générique

La conception générique vise à la spécification de l'architecture technique. Elle permet de dégager les aspects qui ne sont pas fonctionnels mais qui agissent sur ceux-ci.

1.1. Diagramme de déploiement

La Figure 15 est un diagramme de déploiement de notre système. Il fait une représentation de l'architecture matérielle et logicielle du système. Il a les caractéristiques suivantes :

- les différents matériels sont matérialisés par des nœuds ;
- un lien de dépendance représente un lien de communication ;
- il donne la structure d'une plateforme physique ;
- il permet de préciser où se trouvent les processus ;
- il montre comment les objets se créent et se déplacent dans une architecture distribuée.

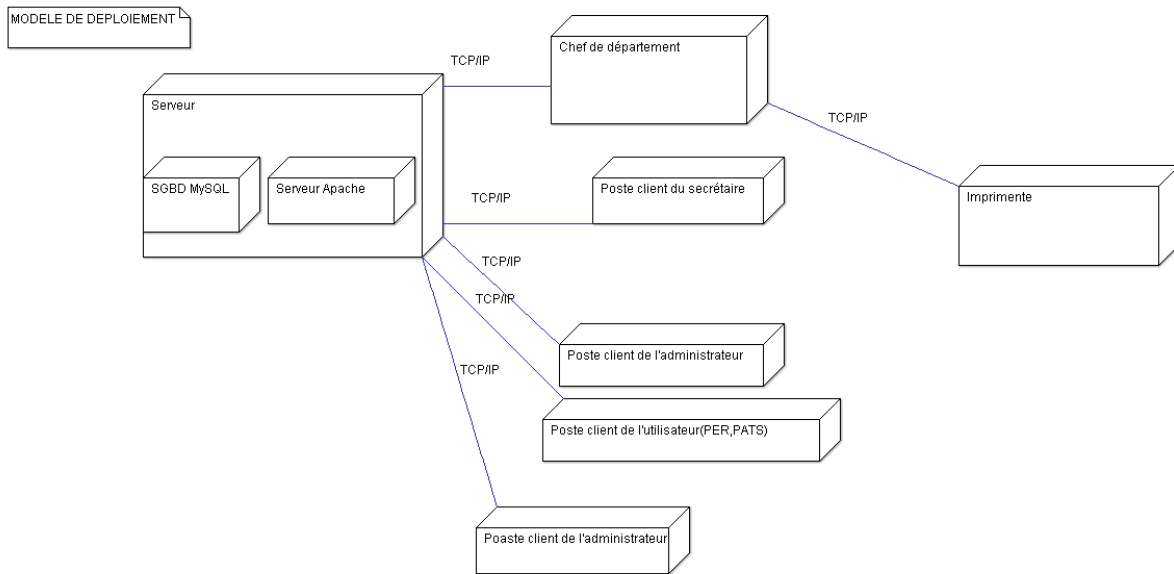


Figure 15: Diagramme de déploiement du système

2. Conception préliminaire

Cette étape est la plus importante du processus 2TUP vu qu'elle représente le cœur. En effet dans cette étape nous allons enfin quitter les deux branches droite et gauche afin de faire la fusion entre les deux études technique et fonctionnelle. Cependant, nous allons développer les catégories d'analyse en couches logicielles conformément au modèle retenu de la solution technique tout en restant le plus indépendant possible des outils de développement. Afin d'aboutir à cela, il est obligatoire de proposer un diagramme de composant.

2.1. Diagramme de composants du système

Un composant est un élément physique qui représente une partie implémentée d'un système. Le diagramme de composants décrit l'organisation du système du point de vue des éléments logiciels comme les modules (paquetages, fichiers sources, bibliothèques, exécutables), des données (fichiers, bases de données) ou encore d'éléments de configuration (paramètres, scripts, fichiers de commandes). Ce diagramme permet de mettre en évidence les dépendances entre les composants (qui utilisent quoi). La [Figure 16](#) suivante représente le diagramme de composant du système.

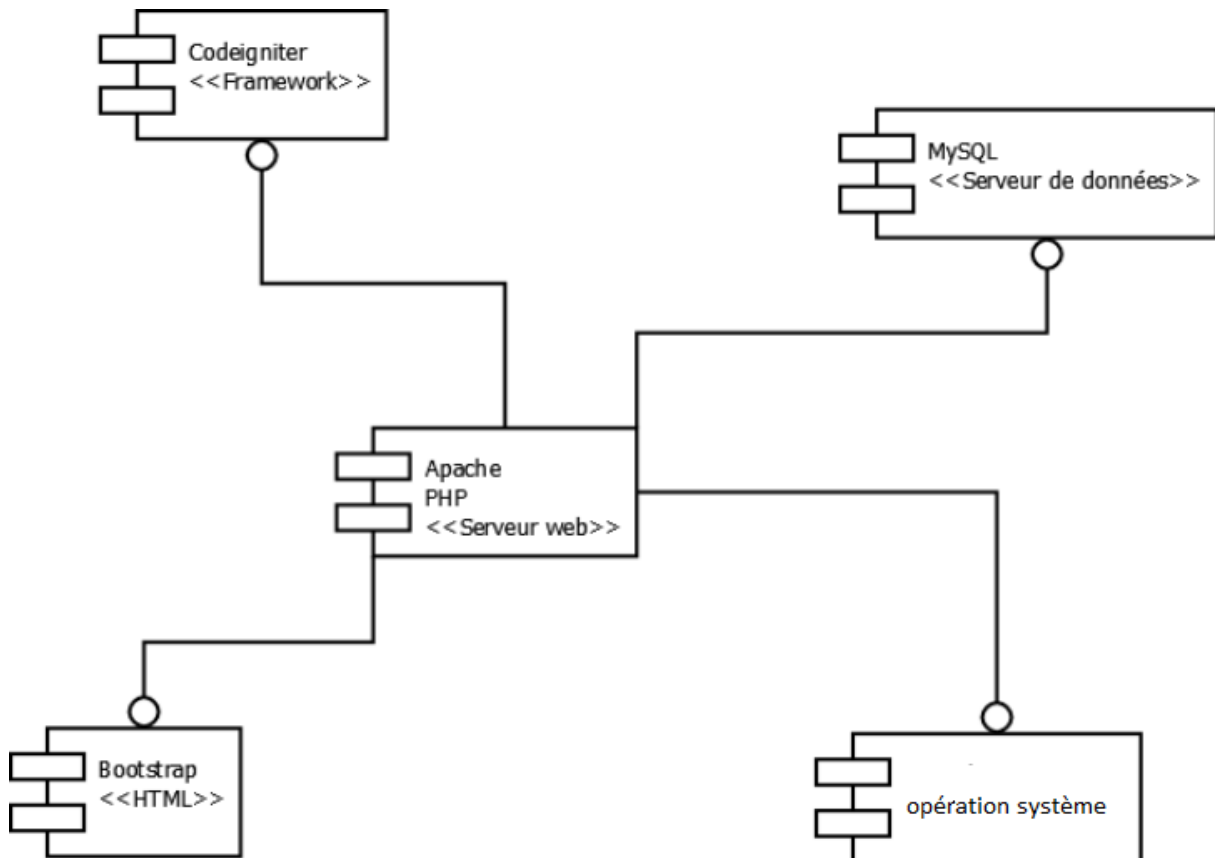


Figure 16: Diagramme de composant du système

Le modèle de conception système organise le système en composants délivrant les services techniques et fonctionnels.

3. Conception détaillée du système

La conception détaillée est une activité qui s'inscrit dans l'organisation définie par la conception préliminaire. Le modèle logique y est particulièrement important dans la mesure où c'est en conception détaillée que l'on génère le plus gros volume d'informations. Dans cette partie, le dictionnaire de données et le diagramme de classes doivent être fournis[7].

3.1. Dictionnaire de données

Le dictionnaire de données est un document qui permet de recenser, de collecter et de trier toute les informations (données) collectées lors des entretiens ou de l'étude des documents[14]. Le **Tableau 9** suivant représente le dictionnaire de données de notre travail. Il permet d'avoir une vue plus claire sur les classes participantes aux fonctionnalités du système.

Classe	Attribut (type)	Opération
User	id_user (int) matricule (varchar) actif (bit) adresse (varchar) email (varchar) fonction (varchar) nom (varchar) pass (varchar) prenom (varchar) profil (int) profilsc (varchar) tel (varchar) departement (varchar) etablissement (varchar) grade (varchar)	Ajouter un utilisateur Modifier un utilisateur Supprimer un utilisateur Afficher un utilisateur
Demandaa	id_demande (bigint) date_dep (date) date_ret (date) datedepot (date) etat (varchar) lieu (varchar) objet (varchar)	Ajouter une demande d'autorisation d'absence Supprimer une demande d'autorisation d'absence Modifier une demande d'autorisation d'absence valide/rejeter une demande d'autorisation d'absence

	<p>organ_frais (varchar)</p> <p>id_user (bigint)</p>	<p>Imprimer une demande d'autorisation d'absence</p>
Demandes	<p>id_demandes (bigint)</p> <p>autresubvention (double)</p> <p>budget (double)</p> <p>date_dep (date)</p> <p>date_ret (date)</p> <p>datedepot (datetime)</p> <p>destination (varchar)</p> <p>etat (varchar)</p> <p>objet (varchar)</p> <p>subvention_dep (double)</p> <p>subvention_rect (double)</p> <p>subventionufr` double</p> <p>type_manif (varchar)</p> <p>id_user (bigint)</p>	<p>Ajouter une demande de subvention</p> <p>Supprimer une demande de subvention</p> <p>Modifier une demande de subvention valide/rejeter une demande de subvention</p> <p>Imprimer une demande de subvention</p>
Mission	<p>id_mission (bigint)</p> <p>date_depm (date)</p> <p>date_retm (date)</p> <p>datedepotm (date)</p> <p>etatm (varchar)</p> <p>lieu_rendu (varchar)</p> <p>moyen_transport (varchar)</p>	<p>Ajouter une demande d'ordre de mission</p> <p>Modifier une demande d'ordre de mission</p> <p>Supprimer une demande d'ordre de mission</p>

	nbjrs (int) objet_mission (varchar) type_frais (varchar) type_indemnité (varchar) motivation (varchar) id_user (bigint)	Imprimer une demande d'ordre de mission
Materiel	id_materiel (bigint) code_barres (varchar) etat (varchar) designation (varchar) annee_bud (int) designation (varchar) nature_unite (varchar) type (varchar) livraison(Date) nombre (int) etat (varchar) id_user (bigint)	Ajouter un matériel Modifier l'état d'un matériel Supprimer un matériel Afficher la liste des matériels

Tableau 9: Dictionnaire de données

Ce dictionnaire nous a permis d'avoir les diagrammes de classe des fonctionnalités les plus importantes.

3.2. Diagrammes de classes

Un diagramme de classes décrit la structure statique d'un système avec des classes, des packages et des relations. Le principe est une représentation des classes d'un programme, de ses membres (attributs) et de ses interactions (relations). Dans ce chapitre nous donnerons le diagramme de classes de l'authentification avant de donner le diagramme de classes des fonctionnalités du reste du système[7].

3.2.1. Diagramme de classes concernant la gestion des utilisateurs et leurs profils

Les classes qui interviennent dans ce diagramme représenté par la **Figure 17** sont : Utilisateur, Administrateur, Assistante, chef de département et espace de travail. Un utilisateur est assistante, chef département, enseignant, ou administrateur et accède à un seul espace de travail. Et chaque espace appartient à un et un seul utilisateur.

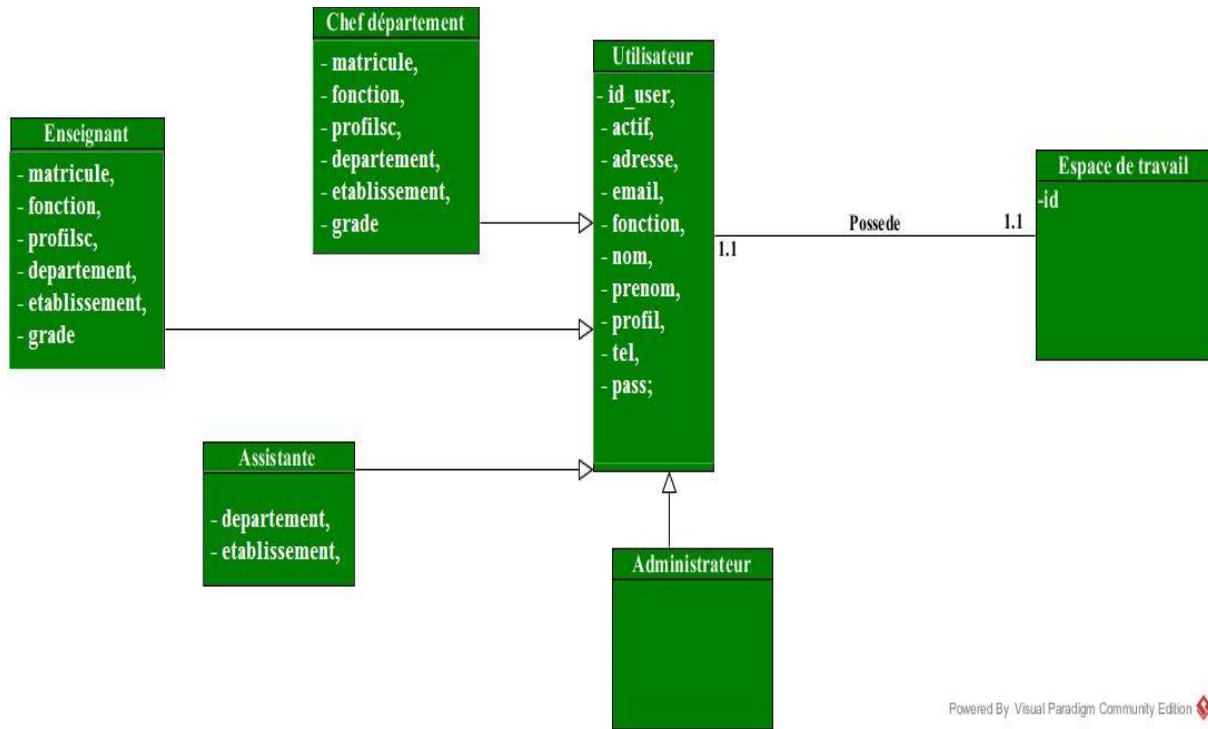


Figure 17: Diagramme de classe concernant les utilisateurs et leur espace de travail

3.2.2. Diagramme de classes concernant le reste du système

Les classes qui interviennent dans ce diagramme de classes représenté par la **Figure 18** sont : Assistante, chef de département, enseignant, demande, matériel.

Une demande peut être une demande d'autorisation d'absence, de subvention, un ordre de mission.

Un enseignant peut faire 0 ou plusieurs demandes (absence) mais une demande n'est faite que par un enseignant ou un chef de département ou son assistante. Le chef de département valide ou rejette la demande, il peut aussi imprimer ou laisser l'assistante le faire à sa place. Un matériel est affecté à un enseignant.

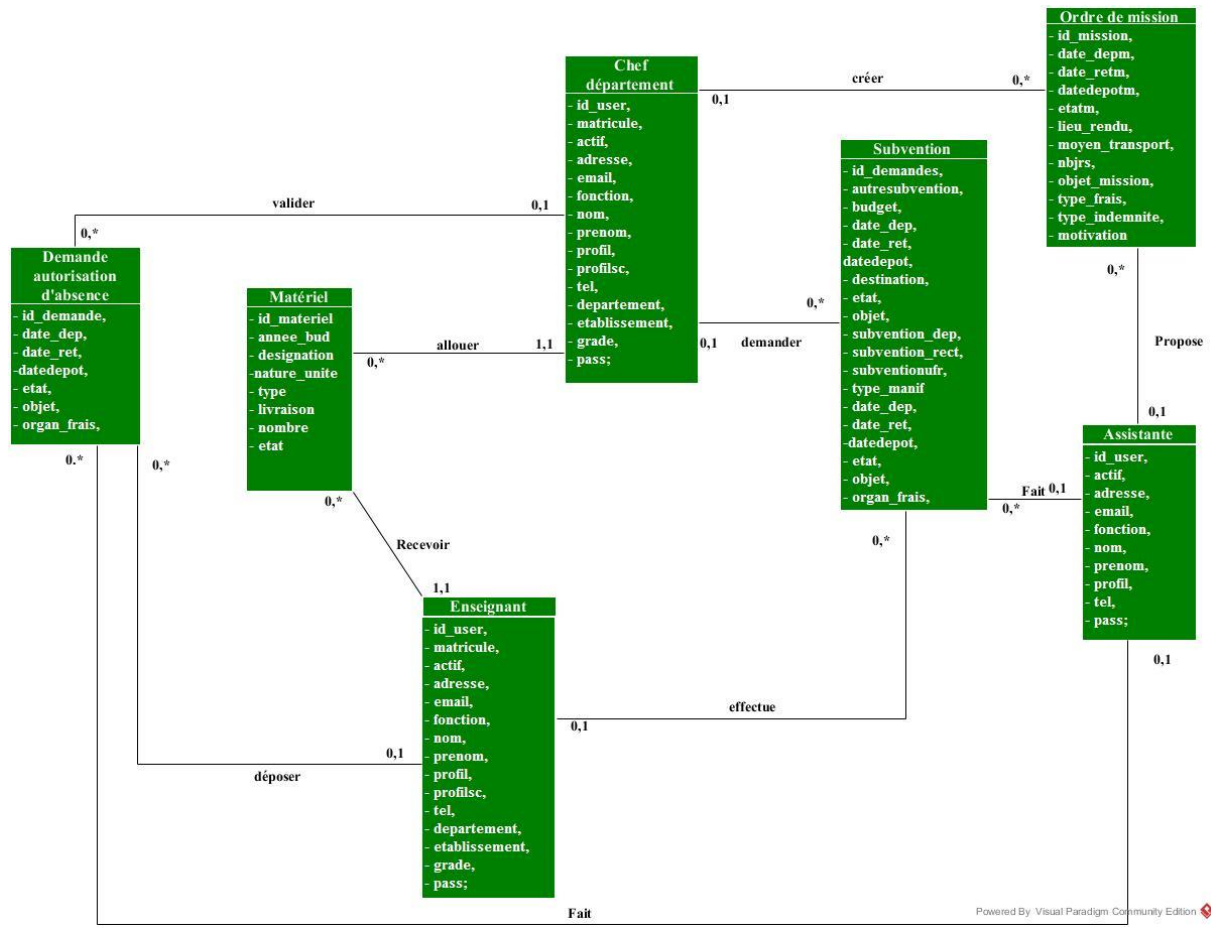


Figure 18: Diagramme de classes concernant le reste du système

Conclusion

Ce chapitre a permis de voir plus clair sur la gestion du département en détaillant les données utiles avec les diagrammes de composants et de classes. Il a également permis d’entamer le dernier chapitre qui porte sur l’implémentation et la présentation du système de gestion du département.

*Chapitre V : Implémentation et
Présentation du système*

Ce chapitre est consacré à la concrétisation du système « gestion du département d'informatique». Ainsi il est question d'installer, de configurer les outils et de coder l'application. La configuration du serveur est opérée ainsi que le codage de l'application qui se fait à l'aide d'un éditeur (Sublime Text). Une fois que cela est fait, l'application doit être présentée.

1. Implémentation du système de gestion du département d'informatique

1.1. Implémentation du SGBD(MySQL)

1.1.1. Le modèle physique de données(MPD)

Les classes subissent des transformations qui aboutissent au Modèle Physique de Données(MPD). Ce modèle peut être défini comme le reflet de la base de données car il permet de la construire.

Les attributs de certaines tables (clés primaires) migrent vers d'autres. Cette migration dépend des cardinalités qui sont définies. Les attributs migrants constituent des clés étrangères dans les tables d'accueil. La **Figure 19** suivante donne le modèle physique de notre application.

```
USER(Id_user,Matricule,Actif,Adresse,Email,Fonction,Nom,Pass,Prenom,Profil,Profils, Tel,Departement,Etablissement,Grade)
Demandeaa(Id_demande,Date_dep,Date_ret,Datedepot,Etat,Lieu,Objet,Organ_frais,#Id_user)
Demandes(Id_demandes,Autresubvention,Budget,Date_dep,Date_ret,Datedepot, Destination,Etat,Objet,Subvention_dep,Subvention_rect,Subventionufr, Type_manif,#Id_user)
Mission(Id_mission,Date_depm,Date_retm,Datedepotm,Etatm,Lieu_rendu, Moyen_transport,Nbjrs,Objet_mission,Type_frais,Type_indemnite,Motivation,# Id_user)
Materiel(Id_materiel,Annee_bud,Designation,Nature_unite,Type,Livraison,Nombre,Etat,#Id_user)
```

Figure 19: Modèle physique de données du système gestion du département d'informatique

1.1.2. La création de la Base de Données et des tables sous MySQL

Notre base de données ainsi que les tables ont été créées manuellement sans l'usage de la console. La **Figure 20** suivante représente l'image qui permet de créer la base de données :

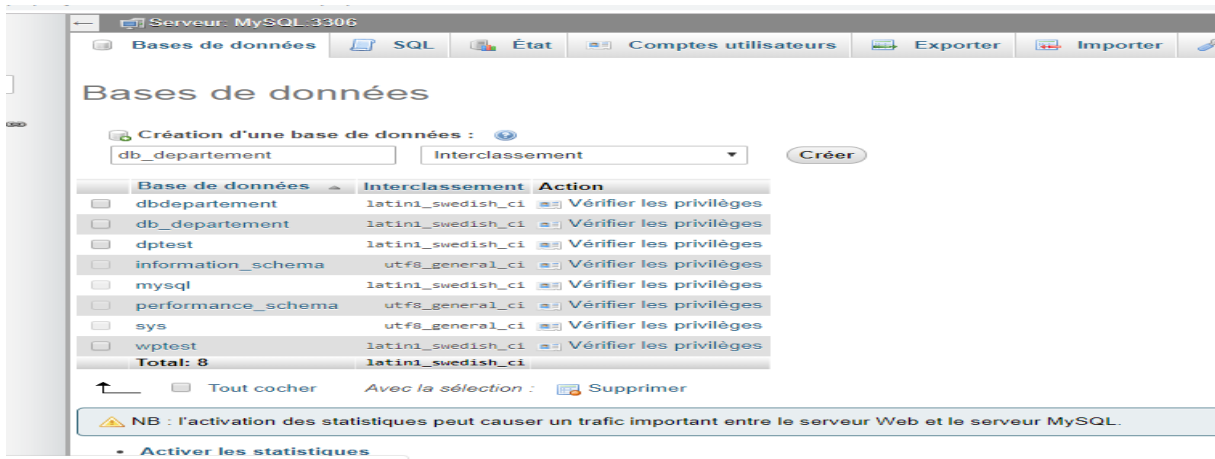


Figure 20: Création de la base de données du système gestion du département d'informatique

Pour ce qui relève des tables, la **Figure 21** représente la création des tables. L'onglet nouvelle table permet de créer une nouvelle table.

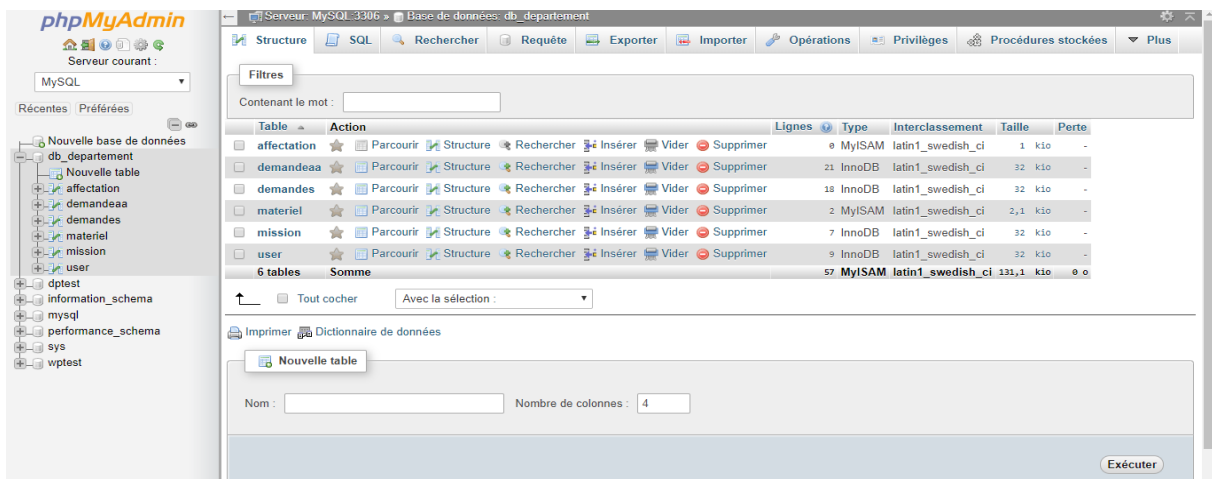


Figure 21: Création des tables de la base de données du système

2. Outils et technologies utilisés

Pour le développement de l'application liée à la gestion du département d'informatique, plusieurs outils sont utilisés :

2.1. Editeur de texte: Sublim Text

Il existe beaucoup de logiciels de traitement de texte comme Word qui eux sont destinés pour l'écriture de lettre ou pour la réalisation de dossier. Sublime Text lui est beaucoup plus l'ami des développeurs. Créé en 2012 par Jon Skinner, Sublime Text est un très bon éditeur

de texte gratuit, capable de s'adapter à l'utilisation de son utilisateur. Il concurrence son ami Notepad++, qui est capable d'en faire autant

Le sublime Text 3 est codé en C++ et Python, disponible sur Windows, Mac et Linux. Le logiciel a été conçu tout d'abord comme une extension pour Vim, riche en fonctionnalités.

Depuis la version 2.0, l'éditeur prend en charge 44 langages de programmation majeurs, tandis que des plugins sont souvent disponibles pour les langages plus rares[15].

2.2. Outil de conception : Visual paradigm

Dans la plupart des logiciels de modélisation UML, l'interface graphique oscille entre gênant et freinant, dans le sens où l'on passe plus de temps à chercher une fonctionnalité ou un panneau de configuration qu'à réellement travailler sur la modélisation en elle-même. L'interface graphique doit faciliter la manipulation des fonctionnalités du logiciel, idéalement s'adapter à la façon de travailler de la personne qui l'utilise. Visual Paradigm a cherché à aller dans cette direction en rendant l'interface graphique intuitive, agréable à utiliser et améliorant globalement l'expérience utilisateur.

Chaque fonctionnalité de Visual Paradigm pour UML est accessible de plusieurs façons. En général, celle-ci est accessible dans le menu déroulant se situant classiquement en haut de la fenêtre. On peut également y accéder par le clic droit qui a l'avantage d'être contextuel et de ne présenter que les fonctionnalités en rapport avec l'objet cliqué. Enfin on pourra se servir de l'interface qui apparaît lorsque l'on sélectionne un élément du diagramme qui permet également d'accéder à ces fonctionnalités.

L'intérêt de ce type d'interface est de permettre à l'utilisateur de rester concentré sur ce qu'il est en train de modéliser en lui donnant accès à tout un tas de fonctionnalités directement en cliquant sur un élément d'un diagramme. Une petite barre de menu apparaît alors, permettant d'accéder aux fonctionnalités relatives à l'élément cliqué. De la même façon, si l'on sélectionne plusieurs éléments, on va avoir accès à des fonctionnalités de groupement, d'alignement et de répartition[16].

2.3. Wamp Serveur

WampServer est une plateforme de développement Web de type WAMP, permettant de faire fonctionner localement (sans avoir à se connecter à un serveur externe) des scripts PHP. WampServer n'est pas en soi un logiciel, mais un environnement comprenant trois serveurs (Apache, MySQL et Maria DB), un interpréteur de script (PHP), ainsi que PHPMyAdmin pour l'administration Web des bases MySQL.

La grande nouveauté de WampServer 3 réside dans la possibilité d'y installer et d'utiliser n'importe quelle version de PHP, Apache, MySQL ou Maria DB en un clic. Ainsi, chaque développeur peut reproduire fidèlement son serveur de production sur sa machine locale[17].

2.4 SGBDR MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels. Il est en concurrence avec Oracle, PostgreSQL et Microsoft SQL Server.

MySQL est robuste et puissant, aux fonctionnalités riches et avancées, capable de manipuler en toute fiabilité de gros volumes de données, mêmes dans des situations critiques.

Un SGBDR est un ensemble de logiciels chargé des fonctions minimales suivantes :

- ✓ le maintien de la cohérence des données entre elles,
- ✓ le contrôle d'intégrités des données entre elles
- ✓ les opérations classiques sur les données (consultation, insertion, modification, Suppression),
- ✓ Les autorisations d'accès aux données [18].

2.5. Outil de programmation PHP orienté Objet

Vous connaissez le PHP mais vous avez le sentiment que votre code pourrait être mieux structuré ? C'est qu'il est peut-être temps de passer à la programmation orienté objet, qu'on appelle aussi POO.

La programmation orientée objet est un nouveau moyen de penser votre code. C'est une conception inventée dans les années 1970, qui prend de plus en plus de place aujourd'hui. Ce paradigme permet une organisation plus cohérente de vos projets, une maintenance facilitée et une distribution de votre code plus aisée![18]

Vous aurez en annexes les [Liens d'installation](#) servant à l'installation de tous ces outils.

3. Présentation de l'application

L'implémentation a permis d'asseoir les bases de l'application et de pouvoir la réaliser. A cette étape, toutes les installations et configurations des outils sont opérées de même que le codage. Au sortir de celle-ci, l'AGADI-UASZ est prête à être utilisée. Elle doit être présentée. Ainsi, à travers les fonctionnalités que propose cette application, le chef de département est en mesure de gérer son département sans difficultés majeures. Quelques interfaces de cette application sont illustrées par des captures d'écrans et des explications.

3.1. Présentation de l'interface de connexion

Cette interface permet aux différents utilisateurs de se connecter c'est-à-dire d'accéder à l'application. Si l'utilisateur donne un login ou un mot de passe incorrect, un message d'erreur lui est envoyé. Il ne peut se connecter que s'il donne un nom d'utilisateur et un mot de passe correcte. La [Figure 22](#) représente la page de connexion.

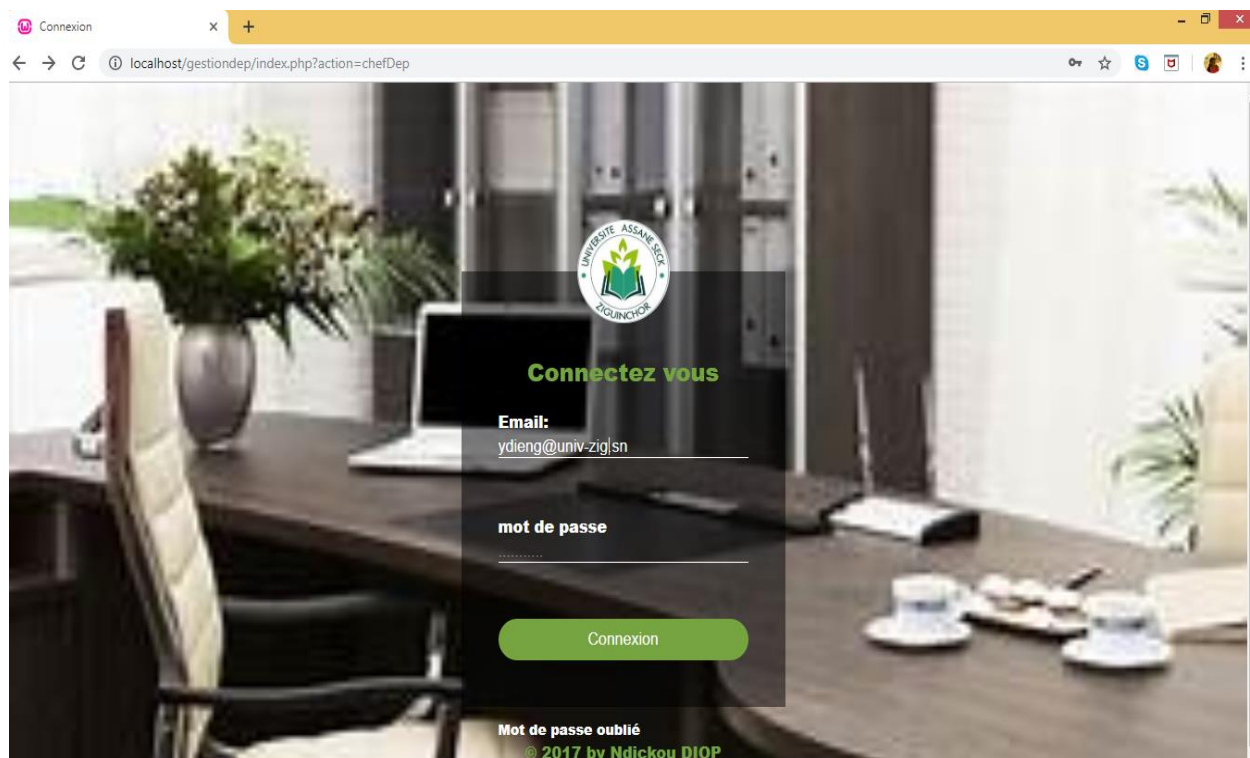


Figure 22: Page de connexion

3.2. Présentation des pages d'accueil du système

3.2.1. Présentation de la page d'accueil du chef de département

Cette interface constitue la page d'accueil de l'application du chef de département. A travers ces différents onglets, le chef de département peut gérer son département via les onglets tels que déplacement, budget, dotation, personnel et historique. La [Figure 23](#) suivante permet de visualiser la page d'accueil du chef de département.



Figure 23: Page d'accueil du chef de département

3.2.2. Présentation de la page d'accueil de l'enseignant

La page d'accueil de l'enseignant permet à celui-ci de faire une nouvelle demande ou de consulter son tableau de bord. A travers ces différents onglets, l'enseignant peut faire ses propres demandes d'autorisation d'absence, de subvention ou de mission. Chaque sous onglet correspond à un tableau de bord qui permet à l'enseignant d'avoir le suivi de ses demandes. La [Figure 24](#) suivante permet de visualiser la page d'accueil de l'enseignant.

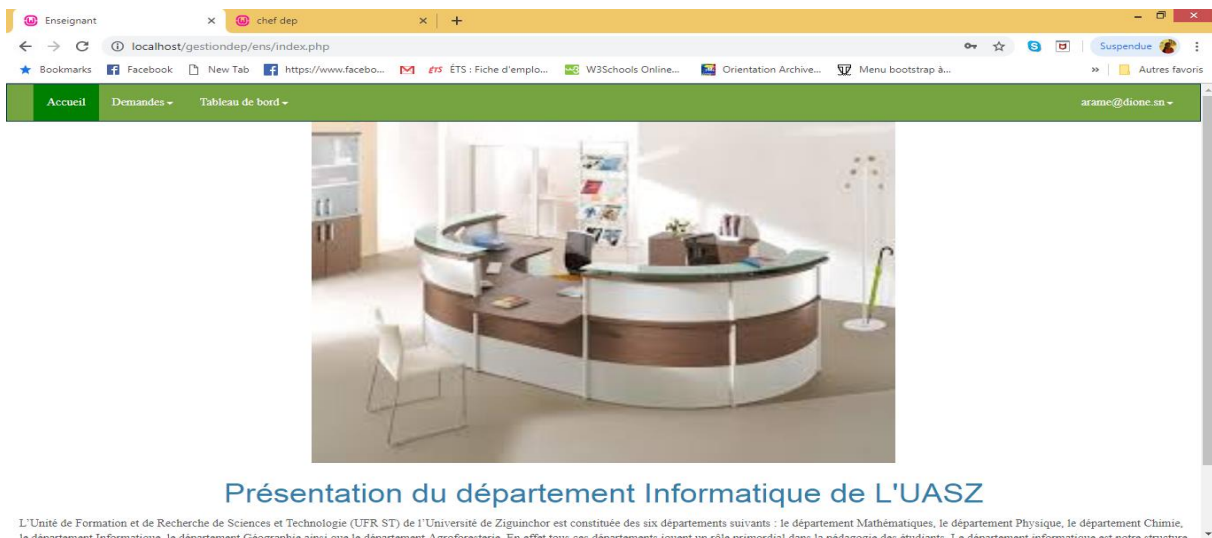


Figure 24: Page d'accueil de l'enseignant

3.2.3. Présentation de la page d'accueil de l'assistante

Cette interface constitue la page d'accueil de l'application de l'assistante du chef de département. A travers ces différents onglets, l'assistante peut aider le chef via les onglets tels que déplacements et budget. La [Figure 25](#) suivante permet de visualiser la page d'accueil de l'assistante.

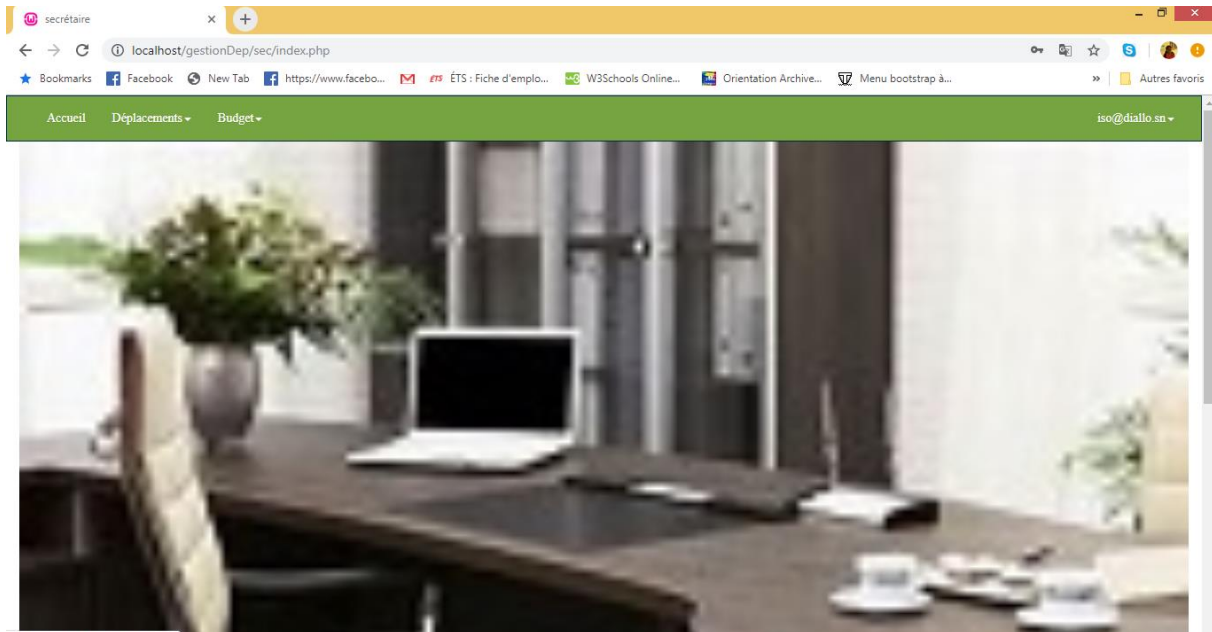


Figure 25: Page d'accueil de l'assistante du chef de département

3.3. Scénario de gestion d'une demande

Ce scénario va décrire les étapes de création d'une demande jusqu'à sa validation ou son rejet. Nous allons simuler, de façon pratique et générale, la gestion des demandes.

Ainsi dans notre cas nous allons simuler la gestion :

- d'une demande d'autorisation d'absence faite par un enseignant ;
- d'une demande de subvention faite par l'assistante pour un enseignant ;
- d'un ordre de mission fait par le chef de département pour un enseignant.

3.3.1. Scénario d'une demande d'autorisation d'absence faite par un enseignant

Une fois connecté, l'enseignant accède à sa page d'accueil clique sur l'onglet « **Demandes** » et choisi « **autorisation d'absence** ».

Une page représentée par la Figure 26 est affichée. Cette page est à moitié remplie car après authentification, le système génère toutes les informations relatives à cet enseignant concernant ce type de demande (Autorisation d'absence, mission ou subvention).

L'enseignant procède au remplissage des champs vide de son formulaire puis clique sur « **Envoyer** »

Ajout d'une nouvelle demande d'Autorisation d'absence

Matricule: 003 Lieu: Lieu du déplacement

Nom: dione Objet: Enter l'objet

Prénom: Arame Date départ: jj/mm/aaaa

Fonction: enseignant Date retour: jj/mm/aaaa

Etablissement: Sciences et technologie Date dépôt: 15/04/2019

Département: Informatique OCF: Organisation Chargée des Frais

Fermer Envoyer

Footer Text

Figure 26: Création d'une nouvelle demande d'autorisation d'absence

Le chef de département à son niveau, se connecte, clique sur l'onglet « déplacement » puis sur « Autorisation d'absence » et il a la liste des autorisations d'absences qu'il peut valider, rejeter, supprimer, imprimer ou visualiser. La [Figure 27](#) permet de visualiser ce phénomène.

Liste des demandes d'autorisation d'absence

Num id	date dépôt	date départ	date retour	etat	Lieu	organ_frais
133	2019-04-11	2019-04-12	2019-04-26	en cours...	mbour	personnel
134	2019-04-11	2019-04-12	2019-04-28	en cours...	kolda	uaszs
136	2019-04-11	2019-04-20	2019-04-27	en cours...	Maroc	personnel

Showing 1 to 3 of 3 entries

Liste des absences

Footer Text

Figure 27: Page de gestion des demandes d'absence

Le chef de département peut aussi afficher les demandes d'autorisation d'absence qui ont été validées que vous pouvez voir à l'[Annexe 7](#).

3.3.2. Scénario d'une demande de subvention faite par l'assistante pour un enseignant

Après connexion, l'assistante décide de créer une demande de subvention pour un enseignant. Elle clique sur « **budget** » puis sur l'onglet « **subvention** ».

L'onglet budget est composé de trois sous onglets qui sont :

- l'élaboration du budget qui est un tableau comportant les sections investissements et fonctionnement au niveau du département.
- Les commandes au niveau du département qui sont représentées ici par un tableau.
- Les subventions où l'assistante ou le chef de département ont la possibilité de créer une nouvelle demande, de modifier, supprimer ou afficher des demandes de subvention.

La figure ci-dessous représente l'onglet ajout de la demande de subvention.

Figure 28: Ajout d'une nouvelle demande de subvention

Ici aussi en insérant le matricule de l'enseignant, le système génère les données relatives à celui-ci. L'assistante va juste compléter les champs vide puis cliquer sur « envoyer ».

Une fois arrivé au chef de département, il procède de la même manière qu'avec les autorisations d'absence pour sa validation ou son rejet.

3.3.3. Scénario d'un ordre de mission fait par le chef de département pour un enseignant

En accédant à sa page d'accueil, le chef de département peut vouloir donner un ordre de mission à un enseignant. IL va dans l'onglet « **Déplacements** » puis clique sur « **mission** ».

Cet onglet permet de faire une nouvelle demande d'ordre de mission, d'afficher, de modifier, supprimer, valider, rejeter ou envoyer des ordres de mission au supérieur. Il peut aussi voir les

détails de la demande. Il permet aussi l'affichage des demandes validées après envoi aux hiérarchies. La [Figure 29](#) permet de visualiser l'onglet mission.

ID demande	date dépôt	date départ	date retour	etat	Lieu rendu	Nombre de jours	type frais
55	2019-04-11	2019-04-13	2019-04-21	en cours...	kaolock	3	frais de séjour
56	2019-04-11	2019-04-20	2019-04-30	en cours...	France	12	frais de séjour
57	2019-04-11	2019-04-20	2019-04-30	en cours...	France	12	frais de séjour
58	2019-04-11	2019-04-20	2019-04-30	en cours...	France	12	frais de séjour

Figure 29: Page de la gestion des ordres de mission

Le reste des onglets représenté par l'[Annexe 8](#), l'[Annexe 9](#) et l'[Annexe 10](#) sont les dotations qui gèrent les matériels du département, l'onglet personnel qui affiche l'ensemble des utilisateurs du système et l'onglet historiques qui relate toutes les activités du département en termes de déplacement, de subvention, de matériels....

Conclusion

Ce chapitre a permis de voir concrètement les fonctionnalités de notre système. Les figures présentées ne sont pas exhaustives. Même si ces fonctionnalités doivent être améliorées là où nous sommes, elles permettent au chef de département de mieux gérer celui-ci et au enseignant d'interagir avec ce chef sans difficulté majeure.

Conclusion générale et perspectives

➤ La mission que nous nous étions fixés est certes de mettre en place une application permettant de gérer le département d'informatique mais elle est valable pour la gestion de tout département universitaire. Nous avons proposé une application d'informatique fiable et optimale permettant au chef de département de dématérialiser la gestion de son département. Cette gestion se fait via notre application grâce à un menu permettant de gérer :

- les autorisations d'absence ;
- les subventions ;
- les ordres de mission ;
- les matériels du département ;
- le budget ;
- le personnel ;
- l'historique.

Le but de cette application est de faciliter la gestion du département. Pour bien présenter les objectifs et les résultats de ce travail, ce document a été divisé en cinq (5) chapitres. Le premier étant la présentation générale et les quatre autres suivent la logique du processus 2TUP. Nous avons eu à utiliser le modèle UML dans chaque phase dudit processus.

En résumé, grâce à ce système, le chef de département sera plus efficace dans la gestion quotidienne du département. Il pourra aussi faire une gestion automatique de son département. Les enseignants pourront interagir avec lui via l'application grâce à une interface dédiée permettant de faire des demandes en ligne.

➤ En perspective nous projetons de :

- proposer l'application à tous les départements de l'UASZ ;
- créer une base de données propre à tous les départements de l'UASZ.
- créer des alertes pour répondre à des rendez-vous au niveau de chaque département.
- élargir le projet en créant :
 - l'interface du responsable des finances de l'UFR ;
 - l'interface du comptable des matières ;
 - l'interface du RP ;
 - l'interface du CSA ;
 - les interfaces du Directeur d'UFR et de ses membres de direction.

Références

- [1] UASZ, « Manuel de procédures » **09/07/2013** .
- [2] Y. Dieng, « RAPPORT 2016-2017 Département Informatique » .
- [3] « Memoire Online - Conception et développement d'une application mobile de vente flash sous android - Sarra, Romdhane ARFAOUI, BEL HADJ SALAH », *Memoire Online*. [En ligne]. Disponible sur:
https://www.memoireonline.com/11/13/7773/m_Conception-et-developpement-d-une-application-mobile-de-vente-flash-sous-android6.html. [Consulté le: 02-avr-2019].
- [4] Zakaria Bouazza, « Méthodologie 2 Track Unified Process », 14:43:16 UTC.
- [5] M. Bigand, *Conception des systèmes d'information: modélisation des données, études de cas*. Technip, 2006.
- [6] L. AUDIBERT, *UML 2 de l'apprentissage à la pratique*. edition Ellipses.
- [7] P. Roques et F. Vallée, *UML 2 en action: De l'analyse des besoins à la conception*. Eyrolles, 2011.
- [8] T. Bailet, *Architecture logiciel*. edition ENI, 2012.
- [9] « Notions d'architecture 3-tier [Application de bases de données, principes et exemples avec LAPP] ». [En ligne]. Disponible sur: <https://stph.scenari-community.org/bdd/lap2/co/webUC003archi.html>. [Consulté le: 11-févr-2019].
- [10] P. Pratt, *Initiation à SQL : cours et exercices corrigés*. Eyrolles, 2001.
- [11] F.-X. Bois, *PHP5 le guide complet*, 1ère édition. 2009.
- [12] J. THOORENS, « 2TUP_2_CapBes ». 2009-2008.
- [13] « Memoire Online - Conception et mise en place d'un logiciel de gestion commerciale au sein d'une entreprise commerciale: cas de établissement KAS Sondirya - Chirac Ruhamy Mulindwa », *Memoire Online*. [En ligne]. Disponible sur:
https://www.memoireonline.com/03/12/5474/m_Conception-et-mise-en-place-dun-logiciel-de-gestion-commerciale-au-sein-dune-entreprise-commer3.html. [Consulté le: 20-févr-2019].
- [14] J. L. Baptiste, *Merise Guide pratique: Modélisation des données et des traitements, langage SQL*. Ed. ENI, 2009.
- [15] HADOUELHADJ Abdelmalek- BERRABAH Abdeslam, « Conception et Réalisation d'une Application de Gestion du Parc Automobile », 27-sept-2017. [En ligne]. Disponible sur: <http://dlibrary.univ->

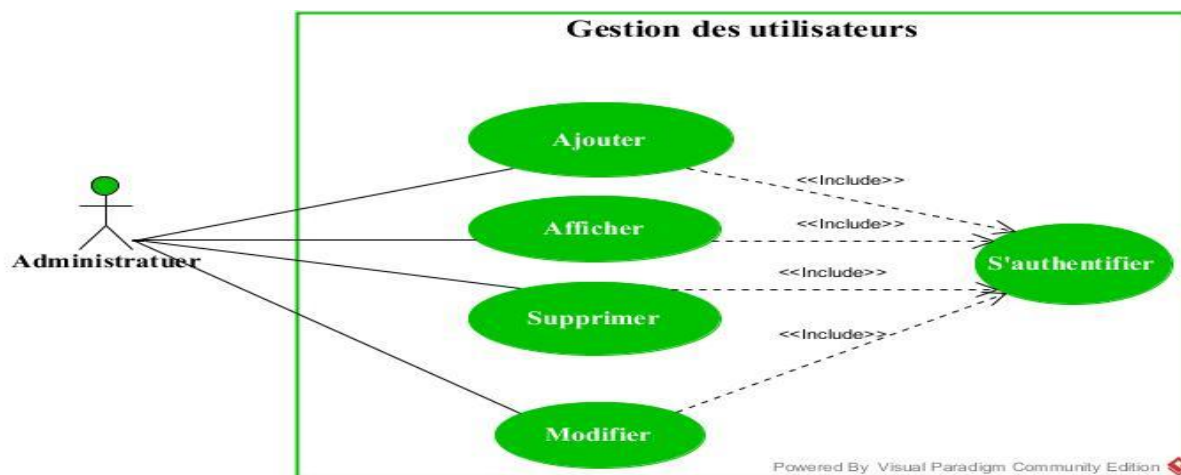
boumerdes.dz:8080/bitstream/123456789/4137/1/HADOUELHADJ%20Abdelmalek%20
OBERRABAH%20Abdeslam.pdf. [Consulté le: 11-avr-2019].

[16] S. Oscar, *Visual Paradigm for Uml*. International Book Market Service Limited, 2013.

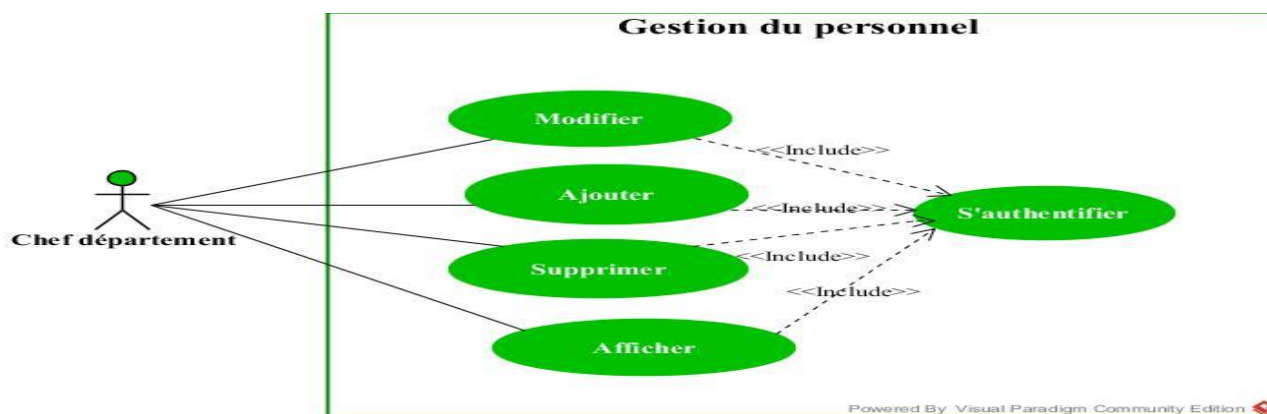
[17] L. Groupe, *Wamp: Comparaison de Wamp, Easyphp, Xampp, Wampserver, Uwamp, Vertrigoserv, Fvpatwds, Mov'amp*. General Books LLC, 2010.

[18] V. Thuillier, *Programmez en orienté objet en PHP*. Eyrolles, 2017.

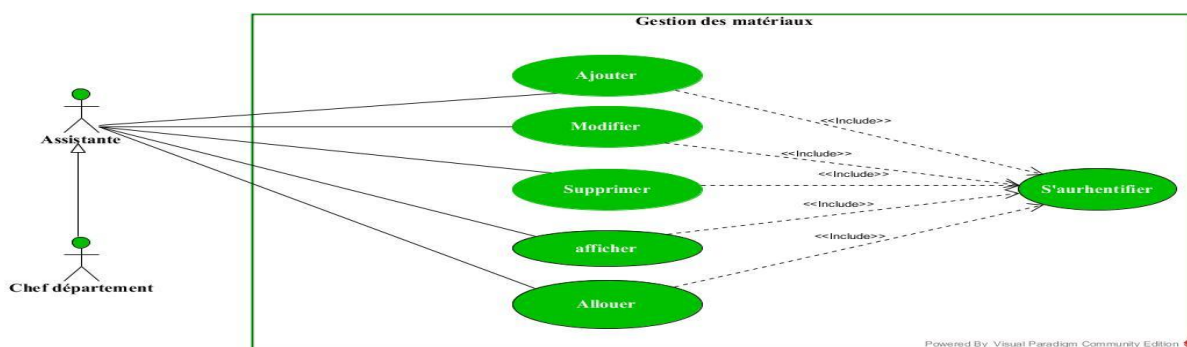
Annexes



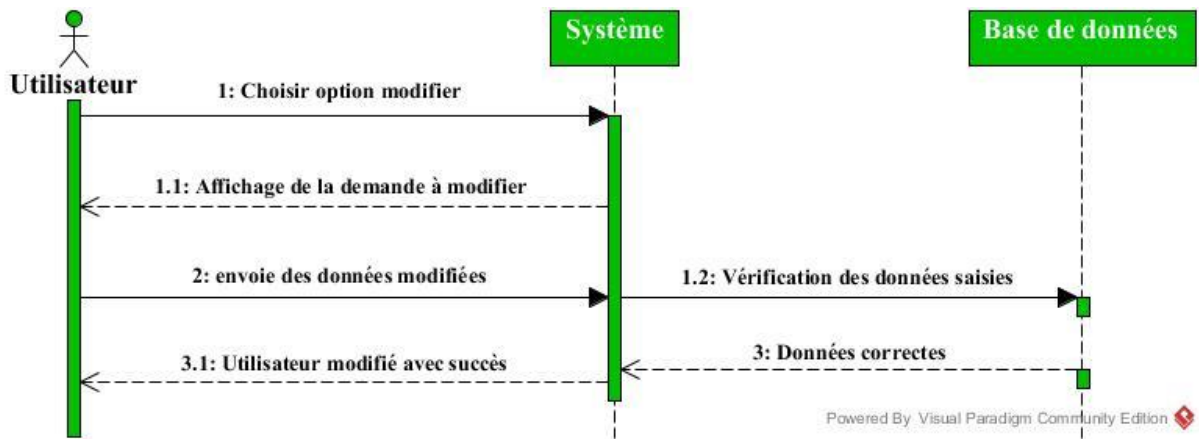
Annexe 1: Diagramme de cas d'utilisation de la gestion des utilisateurs



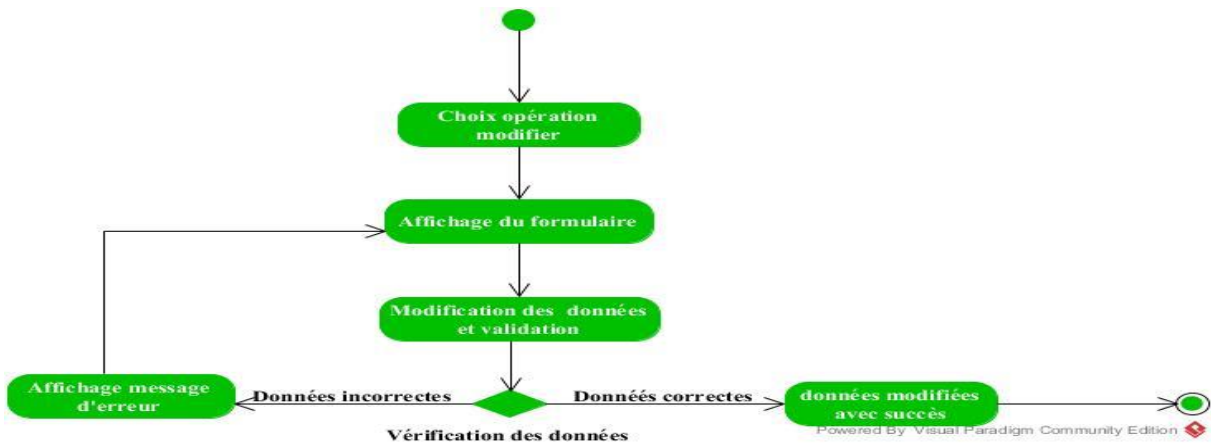
Annexe 2: Diagramme de cas d'utilisation pour la gestion du personnel



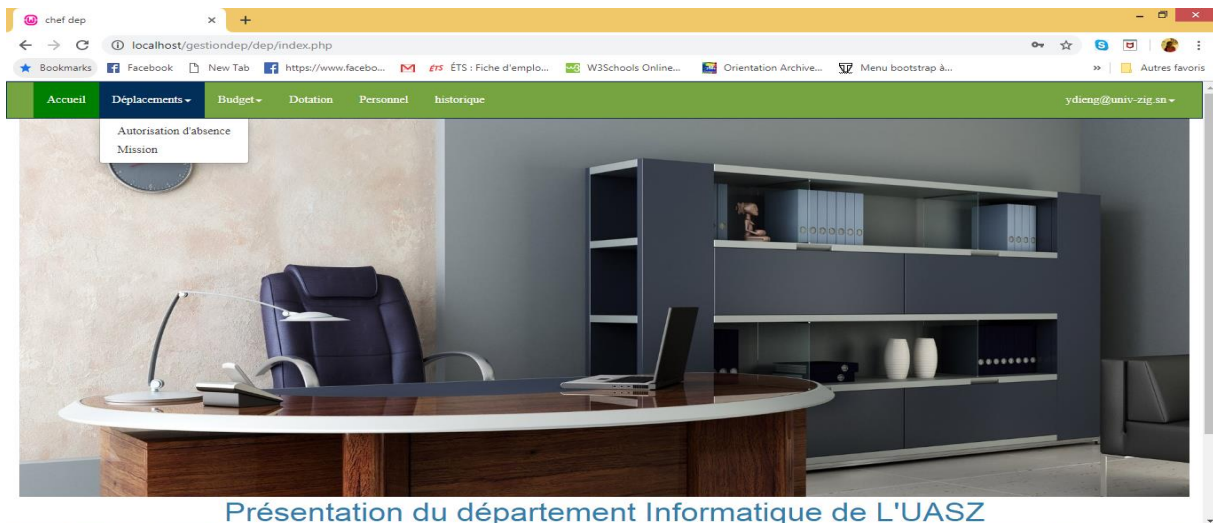
Annexe 3: Diagramme de cas d'utilisation de la gestion des matériels



Annexe 4: Diagramme de séquence du cas « modifier utilisateur »



Annexe 5: Diagramme d'activité modifier utilisateur



Présentation du département Informatique de L'UASZ

Annexe 6: Page de l'onglet « déplacement »

chef dep

localhost/gestiondep/dep/suiviAA.php?val=137

Accueil Déplacements Budget Dotation Personnel historique ydieng@univ-zig.sn

Liste des demandes

Liste des autorisations d'absence

Show 10 entries Search:

Num id	date dépôt	date départ	date retour	etat	Lieu	Objet	organ_frais	id user
135	2019-04-11	2019-04-12	2019-04-28	Validée	kolda	CNRIA	uasz	16
137	2019-04-11	2019-04-20	2019-04-27	Validée	Maroc	reunion	personnel	2

Showing 1 to 2 of 2 entries Previous 1 Next

Footer Text

Annexe 7: Liste des absences

chef dep

localhost/gestiondep/dep/suiviMA.php

Accueil Déplacements Budget Dotation Personnel historique ydieng@univ-zig.sn

Liste des matériels

Ajouter

Show 10 entries Search:

Num id	désignation	Annee_bud	type de matériel	Date Livraison	Nombre	Etat	Bénéficiaire
35	ordinateur portable	2019	matériel de bureau	2019-02-14	1	Excellent	Bouba Ndiaye
37	ordinateur portable	2019	matériel de bureau	2019-02-14	1	Excellent	Bouba Ndiaye
38	ordinateur portable	2019	matériel de bureau	2019-02-14	1	Excellent	Bouba Ndiaye
39	ordinateur portable	2019	matériel de bureau	2019-02-14	1	Excellent	Bouba Ndiaye
40	ordinateur portable	2019	matériel de bureau	2019-02-14	1	Excellent	Bouba Ndiaye
41	ordinateur portable	2019	matériel de bureau	2019-02-14	1	Excellent	Bouba Ndiaye

Showing 1 to 6 of 6 entries Previous 1 Next

Annexe 8 : Liste des matériels

chef dep

localhost/gestiondep/dep/historique.php

Accueil Déplacements Budget Dotation Personnel historique ydieng@univ-zig.sn

Historique

Autorisation d'absence

Liste des demandes d'autorisation d'absence

Show 10 entries Search:

Num id	date dépôt	date départ	date retour	etat	Lieu	organ_frais
92	2019-02-14	2019-02-15	2019-02-23	Validée	kolda	personnel2
118	2019-02-14	2019-02-15	2019-02-22	en cours...	kolda	personnel2
132	2019-03-17	2019-03-18	2019-03-21	en cours...	Maroc	personnel

Showing 1 to 3 of 3 entries Previous 1 Next

Subventions

Missions

Matériels

Annexe 9: Page de l'historique

chef dep

localhost/gestiondep/dep/utilisateur.php

Accueil Déplacements Budget Dotation **Personnel** historique ydieng@univ-zig.sn

Liste du personnel

Show 10 entries Search:

Matricule	Prénom	Nom	Adresse	Téléphone	Fonction	Profil	Grade
001	ndickou	diop	zip	2222222	chef dep		Maitre conference
0001	Youssou	DIENG	Ziguinchor	772721136	chef dep	expert en théorie des graphes	Maitre conférence
002	iso	diallo	dakar	5569420	secrétaire		Professeur
003	Arame	dione	zig	7755694	enseignant		Autre
004	Bouba	Ndiaye	Mbour	777195324	enseignant		Maitre conference
005	Bouba	dione	fouta	5569420	enseignant	on	Professeur
007	Bouba	fall	fouta	7771953240	enseignant	Expert en IA	Professeur
009	Bouba adja	fall	ndar	5569420	enseignant	Expert en IA	Professeur
012	Alassane	sene	zig	777239285	enseignant		Maitre conférence

Annexe 10: Liste du personnel

Annexe 11: Liens d'installation

- <http://cndp.fr/crdp-dijon/Installer-et-configurer-Wampserver.html>
- <https://www.sublimetext.com/3>
- <https://www.visual-paradigm.com>

Guide d'installation d'AGADI-UASZ en local

- **Installer wampServer**
- **Copier le dossier de l'application dans C:\wamp\www**
- **Importer la base de données.**